

We willen graag onze excuses aanbieden aan de firma Komin...Doordat de redactie (ik, ron dus) even niet opgelet heeft, is er een vervelende opmerking aan het adres van de firma Komin verschenen in de vorige quasar. De firma Komin meldde ons dan ook direkt, dat hun prijs al lang aangepast is en dat het Tandata modem uit voorraad leverbaar is. Indien u elders lagere prijzen voor artikelen aantreft dan zegt Komin onmiddellijk te volgen.

Het "winkeltje" dat in hetzelfde artikel aangeraden wordt, Dataskip uit Gouda, kan ik na twee bezoeken echter NIET aanbevelen. Nu zeggen afmetingen (2x3 meter) niets. Maar nadat ik mij, nieuwsgierig geworden door het kleine artikeltje hierover, een selko datamanager aangeschaft had tegen een prijs van f160,- en NIET de door een lid gemelde 138,- inclusief software, bleek het geheel niet te werken. Toen ik met een klacht terug ging beweerde men dat alles in orde was; het horloge en de software. Nadat ik me thuis nog eens verdiept had in de geleverde software, bleek de fout hier wel degelijk in te zitten.

In het vervolg zullen kwaliteits-uitspraken van bedrijven voorzigtiger behandeld worden.

INHOUD:

504 - redaktioneel.

505 - twee 'miracles'...

507 - prisoner aktueel.

508 - archive 2.3 twee maal onder de loep.

509 - Bespreking Prof. Astrologer.

510 - ABACUS toepassing

512 - 3 extra funkties.

513 - SEKTOR,SEKTORI

een uitvoerige bespreking van het gebruik van de tape.

518 - Hoe doe je het in Assembly...

520 - BETAALBAAR SPUL

521 - Koper's dilemma. minidirectory (SB)

523 - OP SAFARI DOOR DE QL.....

526 - Nieuws uit Hilversum

527 - Elektuur 68000-reeks besproken.

528 - GST Macro Assembler besproken.

529 - Ingezonden.....

530 - Kopij, HOE I?

Op de valreep...





BOB VISSER, Scheepmakerskade 30,
3011 VX Rotterdam. 010 - 414.3554



LEDENADMINISTRATIE, sekretariaat,
PENNINGMEESTER, nabestellen oude
nummers.



RON DEN BREEMS, Kroonstaddreef 27,
3067 RT Rotterdam. 010 - 455.1234



REDAKTIE, layout en samenstelling Quasar,
VOORZITTER.



KEES VAN DER WAL, Kwekerijstraat 22,
2613 VE Delft.



MACHINETAAL, SUPERBASIC, PASCAL,
HARDWARE-vragen.
TUSSEN 20.30 en 23.30 UUR !:
015 - 120.360



MICHEL & WILLEM SPANJER,
Hortensialaan 11, 3702 VD Zeist.
19.00 - 22.00 uur: 03404 - 20581



**REPARATIES, HARDWARE en
ONDERDELEN.**



ARD JONKER,
020 - 230.795
MACHINETAAL & HARDWARE-vragen.



MARC KOOL,
020 - 429.345
C & MACHINETAAL-vragen.



FRED BRANDS, Bruggemanlanden 25,
7542 CA Enschede. 053 - 77.22.11



& RUUD REDDINGIUS, Ruysdaelstraat
73b,
1071 XB Amsterdam. 020 - 643.544

VERTALEN ARTIKELLEN.

Stichting SIN_QL_AIR,

Rotterdam.
giro: 4597345.



REDAKTIONEEL



Tja, ik kan het niet nalaten
even te melden dat ik de
QL, sin_QL_air, de QUA-
SAR, de redactionele bezig-
heden en het maken van de layout
tegenwoordig zeer regelmatig in de
steek laat om me bezig te houden
met onze pasgeboren zoon. Daar
kan niets tegenop.

Dat neemt echter niet weg dat jullie
alweer voor een prima inhoud heb-
ben gezorgd.

Quill wordt overal goed gebruikt.
Evenals Archive overigens. De afge-
lopen maand kreeg ik om even voor
half twaalf een telefoontje van ie-
mand die me iets wilde vragen over
een procedure in Archive. Vergeef
me als ik niet al te vriendelijk klonk;
23.30 vind ik een beetje te laat...

Weer iemand anders belde me
afgelopen maand om 21.30 en vond
dat zelf nog net kunnen. Stel u ge-
rust; dat is een prima tijd om te bel-
len wat mij betreft.

Enfin, Quill en Archive dus druk in

gebruik. Vandaar misschien dat ik
juist nu iets over Abacus te vertel-
len heb.

Spelletjes, moet ik bekennen, heb-
ben mij nooit zo getrokken en zeker
de laatste tijd ben ik de computer
steeds meer gaan zien als een hulp-
middel, een stuk gereedschap.

Hoewel ik altijd blijf schreeuwen om
kopij van allerlei aard, wilde ik deze
keer eens extra hard schreeuwen
om kopij betreffende toepassingen
waarbij ABACUS gebruikt wordt.
Mijns inziens is dat een ogen-
schijnlijk onontgonnen gebied. Mis-
schien omdat het zich niet makke-
lijk laat omschrijven. Maar een
dump van de gebruikte formules
kan al bruikbaar zijn.....

Een ander programma waar we nooit
wat van horen is EASEL. Is dat nu
werkelijk alleen een pretpakket, of
zijn er mensen die ook daar alge-
meen bruikbare, dus voor andere le-
den interessante, dingen mee
doen?

Laat het ons weten en help andere
leden aan nieuwe mogelijkheden.

Zijn er overigens leden die zo 'uit-
gekeken' zijn op QUILL en/of
ARCHIVE en/of ABACUS en/of
EASEL, dat ze hun telefoonnummer

hierboven beschikbaar willen stel-
len?

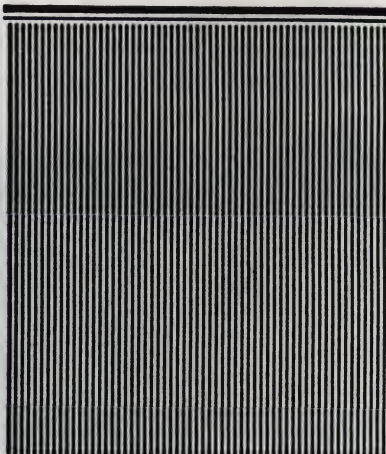
Veel leesplezier en, mocht u de
tocht naar Utrecht maken op 23
mei, dan Tot ziens.

Wees overigens niet ongerust als
uw kopij niet direkt in de
eerstvolgende Quasar staat; dan
komt het zeker in een volgende.

KEEP UP THE GOOD WORK!



Ron.



een circa 3 meter lange kabel, beide eindigend in de inmiddels welbekende Britse telefoonpluggen. In tegenstelling tot vele andere modems die rijkelijk voorzien zijn van schakelaars en LED's een sober aanzien. Het zal duidelijk zijn dat de bediening van het modem met de software gebeurd.

Het aansluiten van het modem is vrij eenvoudig. De korte draad wordt in SER 2 geplugd. De plug aan het lange eind wordt afgeknipt en de draden worden vrijgemaakt. De rode draad van de modem-kabel wordt aangesloten aan punt A van de telefoonplug en de witte aan de B. In het begin zorgde ik er voor dat tijdens het gebruik van het modem geen andere telefoontoestellen aangesloten waren. Naderhand kwam ik er achter dat het contact met bijvoorbeeld Fido niet gestoord werd door een aangesloten telefoontoestel. In de gebruikssituatie ligt het modem achter de QL op de (computer) tafel, wat een goed zicht geeft op de LED.

Van de bijgeleverde cartridge wordt eerst een werkcopy gemaakt, op de zelfde wijze als bij de PSION programma's. Na reset van de QL de werkcopy in de linkerdrive plaatsen en F1 of F2 indrukken. Na enige tijd verschijnt het openingsbeeld en de mededeling dat F3 ingedrukt moet worden. Na dit gedaan te hebben bevindt men zich in de Directory.

Deze Directory biedt plaats aan circa 30 telefoonnummers en is door de programmakers reeds gevuld met de gegevens van de belangrijkste Britse Bulletinboards, die voor ons niet erg bruikbaar zijn. Omdat het modem en Quasar 25 op dezelfde dag in mijn huis arriveerden kon ik putten uit het artikel over het Tandata modem en de telefoonnummers van de daarin genoemde bulletinboards (=BB) overnemen. De Directory toont de naam, het telefoonnummer en het soort bulletinboard. Men heeft de keuze uit Viewdata (=Viditel) of het 'scrolling terminal' type. Een voorbeeld van deze is het HCC Fido net. (de handleiding vermeldt hier overigens dat het een VT52 type is. Heeft dit iets te maken met de CCIT aanbevelingen? Ik heb hierover geen informatie kunnen vinden. Wie?)

In de Edit mode kunnen deze gegevens makkelijk gewijzigd worden, tevens kunnen per BB een viertal veelgebruikte codes vastgelegd worden, die later met een enkele toetsaanslag ingevoerd kunnen worden. Dit is handig om passwords in te voeren.

Het daadwerkelijk opbellen van een BB is zeer eenvoudig. Met behulp van de cursortoetsen wordt het gewenste BB geselecteerd en een druk op de ENTER toets is voldoende om het modem te laten bellen. Het modem is zgn. 'auto dialing'. Erg handig, maar overtuig u er van tevoren van dat wanneer u een nummer voor de eerste keer gebruikt, het inderdaad van een BB is door met een gewoon telefoontoestel het betreffende nummer te draaien. (tenminste..., zolang computers die

opbellen zich nog niet kunnen verontschuldigen voor een verkeerd gekozen nummer)

Er bestaat overigens ook een mogelijkheid om handmatig te bellen, maar dat heb ik nog niet uitgetest. Om een goed contact te krijgen zal, wil men geen onzin op het scherm zien verschijnen, nog wat meer ingesteld dienen te worden. Met behulp van ESC verlaat men de directory en door het opnieuw indrukken van F3 komt men in de command mode. Dit is vergelijkbaar met de PSION programma's. Er wordt een keuze menu getoond, het intypen van de eerste letter van het gewenste commando is voldoende om het commando te activeren.

Overigens is er ook een help toets F1, maar de geboden hulp vindt ik niet erg royaal, zolang de hulp zegt dat je voor meer informatie in de manual moet kijken (in een aantal gevallen)

Met het commando Interface is een aantal zaken in te stellen. De Mode heeft een drietal keuze mogelijkheden. Met Online worden getypte karakters via het serieel interface verzonden. Dit is de normale stand wanneer men communiceert met een BB. Met Local kan men tijdens een sessie opgenomen data met behulp van het Transmit commando weer rustig bekijken. Een derde mogelijkheid is speciaal bedoeld voor communicatie tussen twee QL's.

Met Baud kan de overdrachtssnelheid ingesteld worden. Het heeft geen zin om snelheden in te stellen die het modem niet kan afhandelen. Het modem is geschikt voor 1200/75 baud full duplex en 1200/1200 baud (half duplex?). De pariteit kan ingesteld worden op Even, Odd, Mark en Space. Dit is niet voor ieder BB gelijk. Wie kan mij over de betekenis hiervan eens wat meer uitleggen? Als laatste kan gekozen worden tussen SER1 of SER2, al of niet met handshake-sigitaal.

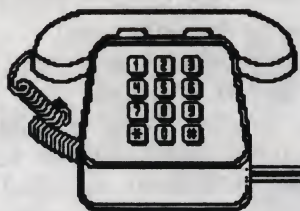
Een commando dat ook in de gaten gehouden moet worden voordat men begint is Width. Men heeft de keuze uit 40 of 80 karakters per regel. Voorlopig gebruik ik 40 karakters per regel met Wrap. (dat is een automatische carriage return met linefeed)

Een nuttig commando is Log. Hiermee kan men tijdens het contact met een BB alle ontvangen informatie 'opnemen' en later met behulp van het Transmit commando 'terugspelen' (Dit is het beste te vergelijken met een cassette recorder)

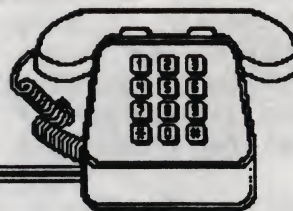
Er bestaan nog twee commando's om ontvangen informatie te behouden. Picture om een scherm vast te leggen en later in Superbasic mbv het LBYTES commando te vertonen. Save om Viewdata pagina's op te slaan. Het Backup commando maakt een nieuwe copy van het gehele programma, met daarin alle (gewijzigde) telefoonnummers, namen etc.

Met het Multitask commando kan men tijdelijk naar Superbasic terugkeren. Dit geeft bij mij nog wel eens problemen.

Het File-transfer commando is alleen bruikbaar bij communicatie tussen 2 QL's, die in ieder geval het



MODEM'S MEN



MIRACLE MODEMS

De 'miracles' zijn de wereld nog niet uit en Kees heeft er nog een paar te koop ook!

Sinds kort heb ik een Miracle modem. Een verbazingwekkend klein kastje (5x2.5x7 centimeter) waar "het" inzit. Aan een kant steek je een stekker in de SER2 poort en aan de andere kant zit zo'n 3 meter snoer wat in het stopcontact van de telefoon komt. Je start de bijgeleverde software op en er wordt een lijst met telefoonnummers voorgeschoteld. Met de cursortoetsen zoek je het gewenste nummer op en je drukt op Enter. Het ding draait dan zelf het nummer; "autodial" heet dat.

Het modem is tamelijk eenvoudig en luistert niet of de centrale kiestoon geeft of dat het nummer in gesprek is. In tegenstelling tot de engelse situatie moeten wij bij het draaien van een interlokaal nummer na het netnummer wachten op de "tweede kiestoon". Bij een ouderwetse centrale wil dat nog wel even op zich laten wachten en het modem draait alweer door ...

Ik raad het dan (voorlopig) ook alleen aan als je op een moderne telefooncentrale bent aangesloten. Ik heb aan Miracle gevraagd of ze een softwareversie kunnen maken die wel een wachtmogelijkheid heeft, maar het kan nog wel even duren voordat ik daar antwoord op heb.

Met het modem kun je terecht bij Viditel en andere databanken die met het 1200/75 baud werken (de CCITT V23 standaard). De meeste FIDO nodes van de HCC kunnen deze standaard ook verwerken.

Volgens de gebruiksaanwijzing moet je er ook files mee kunnen verzenden van de ene QL naar de andere. Dat gaat dan in de 1200 baud halfduplex mode; de een zendt met 1200 en de ander ontvangt alleen maar. Dat heb ik echter nog niet allemaal kunnen uitproberen.

Overigens heb ik nog enkele van deze modems te koop; in de originele verpakking, met software en gebruiksaanwijzing (in 't engels) gebruiksklaar met een nederlandse PTT stekker. De prijs: 185 gulden inclusief verzendkosten.

Kees van der Wal.

Tel: 015 - 120 360 (20.30 - 23.30 uur!)

MIRACLE QL MODEM.

Zijn de wonderen de wereld uit?

Ook Paul Gravink verteld over zijn ervaringen met het Miracle-modem.

Een goed Nederlands spreekwoord zegt wel eens; "Haastige spoed is zelden goed." Al een behoorlijk lange tijd liep ik met het idee rond om een modem aan te schaffen en eens te zien wat er zoal mogelijk is op dit gebied. Tegen de tijd dat ik voor mijzelf een keuze had gemaakt en ik mijn bestelling al had geplaatst las ik in Quasar 24 de oproep om voorlopig geen Miracle modem aan te schaffen omdat men die gezamenlijk wilde gaan inkopen. Goed, inmiddels is het modem gearriveerd en heb ik de werking enigszins onder de knie. Alle mogelijkheden zijn bij lange na nog niet uitgeprobeerd - met name die mogelijkheden die de gebruiker in staat stellen zelf berichten en grafische afbeeldingen op te stellen en het overbrengen van files tussen twee QL's - maar ik meen inmiddels al voldoende ervaringen met het modem te hebben opgedaan om er een artikelje over te kunnen schrijven. Hoewel het een tamelijk lange en uitvoerige beschrijving is meen ik er toch goed aan te doen om het zo te presenteren. Daarvoor kijk ik naar mijn eigen achtergrond. Als een volkomen leek op communicatie gebied met alleen wat theoretische kennis over het principe van een modem en wat ik zoal in de diverse tijdschriften lees. Punt blijft toch dat over een modem lezen en werken met hetzelfde modem twee verschillende zaken zijn.

Je wilt je computer aan de telefoon vastknopen maar je weet niet waar te beginnen. De meeste in de handel zijnde modems zijn alleen te gebruiken met de Modaptor en voorzien vaak niet in de voor de QL benodigde software. Een mogelijkheid is in Quasar 25 al besproken; het Tandata modem. Sinds enige tijd is ook het Miracle QL modem in Nederland verkrijgbaar. Een gunstige bespreking in QL_world deed mij besluiten het aan te schaffen en het maar eens te proberen. Voor f 229,- krijgt men een doosje met daarin een modem, een cartridge met de software en een beschrijving. Om met het laatste te beginnen. Ik kan er nog steeds niet aan wennen dat je bij een Philips citrussers van f 40,- een keurig uitgevoerde beschrijving en gebruiksaanwijzing krijgt en bij veel computerapparatuur een onooglijk, gefotocopieerd boekje dat de gebruiksaanwijzing moet voorstellen.

Het modem zelf bestaat uit een glanzend zwart doosje, ter grootte van twee cartridges maar dan wat dikker. Op de bovenzijde voorzien van een rode 'on line' LED en voorzien van een korte en

Archive nader bekeken.....

ARCHIVE EN MASTERFILE VERGELEKEN

Dat een klein broertje soms(?) sneller kan zijn blijkt uit de vergelijking die Dick voor u bekeek.

Een paar jaar geleden vond ik dat mijn zojuist aangeschafte Spectrum (ook) nuttig gebruikt diende te worden. Waarschijnlijk was het een (eigenlijk onzinnige) reactie op de veel gestelde vraag: "Waar gebruik je die computer nou voor?".

In ieder geval besloot ik mijn uitgebreide muziekverzameling in een database (Masterfile) te stoppen. Vele avonden heb ik besteed aan het intypen van de gegevens, mompelend het toetsenbord van de Spectrum vervloekend. Uiteraard had ik weinig zin al deze gegevens over te gaan typen in Archive, toen ik een QL aangeschaft had. Met een RS-232 verbinding is de conversie van Masterfile naar Archive uiteindelijk gelukt, hoewel het me zeer veel tijd en moeite gekost heeft.

Overigens heb ik het omstreden programma *The Editor* hierbij erg leren waarderen. Te laat kwam ik er via een oproep in Quasar achter dat ik die conversie wel iets makkelijker voor elkaar had kunnen krijgen.

In QL User van december 1985 was over hetzelfde onderwerp reeds een uitgebreid artikel met listings verschenen...

Het aardige was nu dat ik, gebruikmakend van dezelfde data, Masterfile en Archive met elkaar kon vergelijken.

Een eerste opvallend verschil was dat Masterfile, bestemd voor een 48 K computer, de data efficiënter in een file wist op te bergen. Dezelfde data die in Masterfile nog geen 30 K nodig hadden, besloegen in Archive ongeveer 45 K. Anderhalf keer zoveel dus.

Ook op het punt van de (zoek)snelheid sloeg Masterfile bepaald geen slecht figuur. Als je in Archive een bepaald record wilt opzoeken, heb je drie commando's ter beschikking.

Verreweg het snelste commando is LOCATE. In mijn bestand werd elk record binnen een seconde gevonden. LOCATE kent echter twee beperkingen. Ten eerste doorzoekt het alleen het veld dat gebruikt is om de file te sorteren en ten tweede moet de string waarmee je zoekt het begin van de veldinhoud van het gezochte record bevatten. Een voorbeeld kan deze orakeltaal misschien verduidelijken.

Stel dat je een muziekbestand hebt dat gesorteerd is op de naam van de componist. Indien je nu via LOCATE het record met de naam 'Beethoven' wilt vinden, kun je wel LOCATE "Beet" gebruiken, maar niet LOCATE "hoven".

De andere twee zoek-commando's in Archive zijn SEARCH en FIND. Ik had verwacht dat SEARCH van deze twee het snelst zou zijn, daar SEARCH in tegenstelling tot FIND, slechts n (gespecificeerd) veld doorzoekt terwijl met FIND de gehele file doorzocht wordt.

FIND was echter de winnaar: het kostte ongeveer 25 seconden om het juiste record te vinden. Je kunt bij FIND met een string zoeken, zodat je 'Beethoven' kunt vinden met bijvoorbeeld FIND "eetho".

Zoals gezegd was SEARCH de traagste. Bovendien mag je geen strings gebruiken, zodat je verplicht bent de correcte veldinhoud in te vullen. Zoeken met SEARCH componist\$ = "Beethoven" kostte niet minder dan 50 seconden. Nu bestaat er wel een truc om toch met een string te kunnen zoeken. Je maakt dan gebruik van de INSTR functie:

met SEARCH INSTR(componist\$. "eetho") > 0
kostte het eveneens ongeveer 50 seconden. Indien je ook nog de LOWER functie toevoegt om onafhankelijk van grote en kleine letters te kunnen zoeken, moet je in totaal zo'n 75 seconden wachten voor het Beethoven-record gevonden is....

Dit laatste is dan precies de manier waarop bij de Spectrum in Masterfile gezocht wordt: veldspecifiek, met string en onafhankelijk van grote of kleine letters.

Om het beheerst uit te drukken:

Masterfile is sneller.

Het kost de Spectrum slechts 2 seconden.

Alles nog eens op een rijtje:

LOCATE	1 sec.
FIND	25 sec.
SEARCH	50 sec.
SEARCH +	75 sec.
MASTERFILE	2 sec.

De conclusie is in ieder geval dat Archive 2.3 geen snelheidsmonster is en het op dit punt tegen z'n kleine broertje moet afleggen. Daar staat weer tegenover dat Archive veel flexibeler is dan Masterfile en met grotere bestanden kan werken. Maar toch..... waar blijft versie 2.4 ?

Dick Verroen

zelfde communicatieprogramma gebruiken. Met dit commando is het mogelijk om data met 1200 baud te verzenden of te ontvangen. Dit heb ik nog niet kunnen uitproberen. Hoe werkt een en ander in de praktijk? Na het eerste gemodder gelukke het me al vrij snel om contact te krijgen met Viditel, RvV data en Hobbytel. (allen van het Viewdata type) Interface als volgt instellen:

Mode: Online, Baud: 1200M75 of 1200S75, Parity: even en SER2i.

Het contact krijgen met Fido (Rotterdam-1 en Rozenburg-1) was wat moeilijker. Belangrijk is om niet te vergeten dat Fido van het Terminal type is. Interface als volgt instellen:

Mode: Online, Baud 1200M75 of 1200S75, Parity: odd en SER2i. Hoewel Fido uitgebreide helpfaciliteiten biedt is enige voorkennis wel handig. In de catalogus van de HCC dagen 1985 staat een uitgebreide beschrijving van Fido en tegenwoordig geeft de HCC ook een goedkoop boekje uit met veel informatie over Fido.

Het is in het begin aan te raden om elke keer dat men contact heeft met een BB het Log commando in te schakelen, zodat men later precies kan zien wat men heeft gedaan. De cartridge is verder nog voorzien van een demo met daarop Viewdata beelden van een aantal Britse BB's. Ook is een basic programma toegevoegd waarmee men zelf berichten kan opstellen om eventueel te verzenden. In de manual is ook aangegeven hoe men met behulp van Quill berichten voor verzending kan opstellen.

Voor vragen en opmerkingen kunt u me bellen of schrijven (dit laatste bij voorkeur daar ik vaak afwezig ben). Bent u zelf in het bezit van een Miracle modem of in het bezit van een ander modem en geïnteresseerd in communicatie neemt u dan eens contact met mij op.

Hopende een opbouwend steentje aan Quasar bij te hebben bijgedragen,....

Paul Grevink
Rodenrijsestraat 95 B
3037 NE Rotterdam 010 465839



Prisoner's dilemma aktueel.

Paul Grevink signaleerde ook het artikel van Jan Jacobs in de Volkskrant en geeft daarvan hieronder een samenvatting.

De inhoud van het volgende artikel is afkomstig uit de zaterdag bijlage Wetenschap & Samenleving van de Volkskrant van zaterdag 25 april 1987. Na een inleidend stukje waarin de problemen rond het prisoner's dilemma en de vreemde handelspartners uiteen worden gezet (bekende materie voor de regelmatige Quasar lezer)... In 1980 werd een computertoernooi gehouden dat op dezelfde manier was opgezet als nu in de Quasar. Zelfde probleem en dezelfde puntentelling. Alle deelnemende computerprogramma's kwamen elkaar in een willekeurige volgorde meerdere malen tegen. En elk programma sloeg de resultaten van eerdere ontmoetingen op, zodat het van elke tegenstander wist hoe vaak deze de boel bedroog, of juist meewerkte. De deelnemende computerprogramma's moesten proberen uit alle ontmoetingen samen zoveel mogelijk punten te bemachtigen. Winnaar van het toernooi was een 4-regelig programma met de naam TIT FOR TAT, in goed Nederlands; boontje komt om zijn loontje. Het werd ingezonden door Anatol Rapaport van de universiteit van Toronto. De strategie is zo eenvoudig dat je het bijna niet gelooft: laat de eerste keer altijd een volle tas achter, en doe bij alle volgende ontmoetingen precies wat de tegenstander bij de vorige ontmoeting deed. TIT FOR TAT won met glans van slinkse programma's die probeerden uit te vinden wat de zwakste plekken van de tegenstanders waren. En in een later toernooi, waaraan eminente wetenschapsmensen uit verschillende disciplines deelnamen, bleek TIT FOR TAT eveneens de sterkste. In een boek, The Evolution of Cooperation, geeft de politicoloog Robert Axelrod een analyse van het opmerkelijke succes van TIT FOR TAT. Hij doet dat aan de hand van eigenschappen die het programma bezit. TIT FOR TAT is allereerst 'aardig'. Het laat nooit als eerste een lege tas achter. TIT FOR TAT is daarnaast vergevingsgezind. Het programma koestert geen langdurige wrok als de tegenstander een keer een lege tas heeft achtergelaten. Maar het reageert wel meteen met een lege tas, totdat de tegenstander weer meewerkt. En wat misschien het belangrijkste is: andere tegenstanders - zelfs de gemeenste - hebben snel in degaten dat het geen zin heeft TIT FOR TAT dwars te zitten, en passen hun gedrag aan. Zonder deze 'eigenschappen' kan een programma, zegt Axelrod, nooit winnen. In latere toernooien scoorden verbeterde versies van TIT FOR TAT beter dan het origineel. Maar het waren wel aardige, vergevingsgezinde, rechtvaardige en herkenbare programma's.

auteur: Jan Jacobs

huizen, de planeten en de draconis. Een eventuele retrograde-gang van een planeet (de planeet "loopt dan terug") wordt nu niet aangegeven.

Ik denk dat de huizen volgens het placidus-systeem zijn berekend. Ik heb de standen even vergeleken met de standen uit m'n eigen programma en er bleken bij de huizen afwijkingen op treden tot 15 boogminuten toe, alleen het Mc en de Ascendant waren vrij nauwkeurig. De verschillen bij de planeetstanden bedroegen maximaal 2 boogminuten.

Hierna worden, als je ervoor gekozen hebt, de aspecten tussen de planeten berekend. Het programma berekent geen aspecten met de huiscuspen. Je kunt nu de gegevens van de radix saven en een karakterbeschrijving naar het scherm of de printer laten sturen.

Het is volgens mij niet mogelijk om de printer een horoscopetekening te laten printen.

Een andere mogelijkheid, die m.i niet zoveel met astrologie te maken heeft, is die waarbij je het zonnestelsel op de dag van je geboorte in perspectief kunt zien. Je kunt de schaal en de invalshoek veranderen en je kunt de afstand van de aarde tot de verschillende planeten laten berekenen. Toch leuk!

Bij progressieve horoscopen wordt het "1 dag voor 1 jaar" systeem gebruikt. Je moet dan "30" invullen als je de horoscoop van de dag waarop je 30 jaar wordt wilt berekenen.

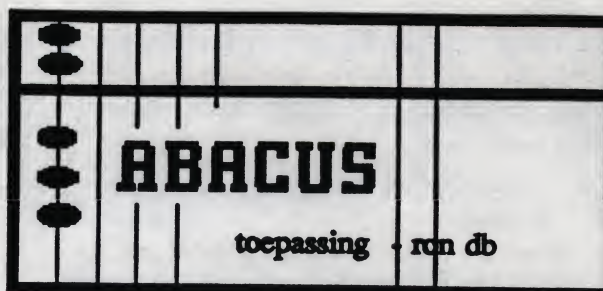
Van deze horoscopen kun je ook weer de aspecten laten berekenen en een tekening laten maken. Ook wordt berekend in welke huizen de planeten zich bevinden en hoe groot de afstand tot die huiscusp is. Een eventuele retrograde-gang van een planeet wordt nu wel aangegeven, evenals het direct gaan lopen van een planeet!

Bij het vergelijken van 2 horoscopen worden de planeetstanden van de 2 horoscopen in een horoscopetekening ingetekend en kun je de aspecten tussen de planeten van de 2 horoscopen laten berekenen. Dit zijn, kort samengevat, de mogelijkheden van het hoofdprogramma.

Met een ander Basic-programma is het mogelijk de karakterset te wijzigen en de gewijzigde karakterset te saven. Met weer een ander Basic-programma is het mogelijk zelf horoscoop-interpretaties te maken waarbij gebruik kan worden gemaakt van de nieuwe karakterset. Deze interpretaties kunnen weer worden gesaved.

Tot zover de bespreking van SUPER-ASTRO-LOGGER.

Hans Brinkmann.



Systematisch Beheer en Onderhoud met ABACUS.

In onderstaand artikel wordt een model voor ABACUS beschreven dat in eerste instantie opgezet is om de onderhoudskosten voor een aantal woningen te kunnen overzien en te verdelen over een aantal jaren. Voordat ik eigenaar werd van een eenvoudige maisonette in een blok van acht, had ik nooit gedacht, dat de kosten voor het onderhoud van een huis zo op konden lopen.

Wellicht heeft u diezelfde gedachte en kunt u met behulp van het hier beschreven model uw kosten wat spreiden.

Met enige fantasie en aanpassingen is het model waarschijnlijk ook te gebruiken voor andere doeleinden; dat ter beoordeling van de altijd kritische lezer.....

Eén van de eerste zaken die mij als voorzitter van een (wettelijk verplichte) vereniging van huiseigenaren te doen stond, was het vinden van een administratiebureau of administrateur. Nu bestaat deze vereniging uit slechts acht personen en één van hen bleek werkzaam bij een woning-coöperatie. Deze buurman vond het zonde om geld uit te geven aan een administratiebureau en meende, gezien zijn werk, voldoende achtergrond te hebben om zo'n taak op zich te nemen. Samen zijn we toen aan het werk getogen om te trachten niet alleen een eenvoudige administratie en huishoudelijk reglement in elkaar te sleutelen, maar om een meer-jaren-plan te formuleren waar we enig houvast aan konden hebben betreffende de te verwachten uitgaven aan onderhoud van onze woningen.

Bij het bedrijf waar mijn buurman werkt wordt gebruik gemaakt van een fors computersysteem om onderhoudsuitgaven te bepalen en plannen. Al pratend hierover stelde ik voor om een poging te wagen dit zelfde te doen op mijn 'persoonlijke computer', toen nog 128K.

"Nou dat lijkt me nogal aan de optimistische kant" zei mijn toch altijd optimistische buurman, een beetje glimlachend kijkend naar een in zijn ogen uit de krachten gegroeide reep chocolade; mijn QL....

Dat weerhield mij er natuurlijk niet van om door te vragen en met enige pret in zijn ogen kwam hij enkele dagen later met een voorbeeld-formulier van het door zijn firma gebruikte systeem. Na enige uitleg beweerde ik dat het waarschijnlijk -je kan nooit voorzichtig genoeg zijn, zeker niet met

Uitbreidingen in ARCHIVE 2.3

Versie 2.3 heeft enkele mogelijkheden t.o.v. zijn voorganger; Peter Roozemaal vertelt er het een en ander over.

Versie 2.3 van Archive heeft ten opzichte van eerdere versies een aantal wijzigingen ondergaan. Er zijn een aantal nieuwe functies toegevoegd, het is mogelijk geworden procedures in object-code (al vertaald) weg te schrijven en de ordening van de karakterset is door elkaar gegooid.

Om met het laatste te beginnen:

het is nog steeds mogelijk om functietoetsen en de cursortoetsen met behulp van 'inkey' en 'getkey' te lezen. De bijbehorende karaktercodes zijn niet de codes die in de tabel in de USER-GUIDE staan maar andere codes. Het is het veiligst om met `print code(getkey())` na te gaan welke code er bij een bepaalde toets hoort. De syntax van de standaardprocedures `save`, `load`, `merge` en `run` is zodanig veranderd dat je door tussenvoegen van het woord `object` (bijv. `save object 'bomen'`) kunt aangeven dat de procedures moeten worden gelezen van dan wel geschreven

naar een objectfile. Hierdoor kunnen procedures sneller geladen en gesaved worden. Met `save protect` kun je je procedures beschermen tegen editen en listen.

Archive kent in de nieuwe versie ook de functies *atn*, *cos*, *deg*, *exp*, *ln*, *pi*, *rad* en *sin* die hetzelfde doen als hun Basic-equivalenten.

Voor conversie zijn de volgende functies:

dec(geval, decimalen, lengte), die *geval* in decimale representatie met *decimalen* cijfers achter de komma in een string ter lengte *lengte* zet;

gen(geval, lengte), die *geval* omzet in een string met lengte *lengte* en

num(geval, lengte), die *geval* omzet in een geheel getal in een string ter lengte *lengte*.

De functie *fieldn(n, [logische filenaam])* geeft de naam van veld *n* van een file.

En de functie *value(string-expressie)* geeft de waarde van de variabele waarvan de naam gegeven wordt in de string-expressie. In de helpfile van ARCHIVE staat extra informatie over object, protect en de nieuwe functies.

Peter Roozemaal

SOFTWARE BESPREKING

SUPER-ASTROLOGER

Ook de mening van Hans Brinkmann kan nuttig zijn als u er over dacht Professional Astrologer te gaan aanschaffen. Hoewel vorige maand reeds bekeken, hier toch nog een kleine bespreking van dit 'hemelse' programma.

Het geheel bestaat uit een aantal Basicprogramma's ;t.w. het hoofdprogramma, een programma om de karakterset te veranderen en een soort van tekstverwerker waarmee je zelf horoscoopinterpretaties kunt maken. Daarnaast worden een file voor de nieuwe karakterset en een QUILL-file meegeleverd. Bovendien zijn er veel files met gegevens van bekende personen. Vlak na het opstarten liep het programma vast met een foutmelding omdat in de Basic-listing de variabele "EX" wordt gebruikt. In ongewijzigde vorm kun je het programma dus niet tegelijkertijd met de TT-Toolkit of de disc-interface van de familie Spanjer gebruiken.

Nadat ik de listing had veranderd liep het verder goed. Nadat ik het hoofdprogramma opnieuw had

gestart werd het mogelijk de printer en de karakterset aan te passen. Je kunt ook de inhoud van je opslagmedium bekijken (makkelijk te veranderen in de Basic-listing) en de gegevens van een bekend persoon laden. Vervolgens moet je je geboorte gegevens invoeren, zoals je geboortedag, je geboortetijd (staat op uittreksel uit het geboorteregister), een tijdcorrectie i.v.m. het verschil tussen de GMT en de plaatselijke tijd en de coördinaten van je geboorteplaats. De geboortetijd kan tot op de minuut nauwkeurig ingevoerd worden, de nauwkeurigheid van de coördinaten is een seconde.

Voor het berekenen zijn er 3 mogelijkheden; je kunt een radix (geboortehoroscoop) of een radix met progressies laten berekenen en je kunt 2 horoscopen laten vergelijken. Daarnaast kan ook gekozen worden voor interpretatie van de aspecten, de planeten in de tekens, de planeten in de huizen en voor een horoscooptekening. Na ongeveer een minuut worden de berekeningen getoond. Als je voor de radix hebt gekozen krijg je een nette horoscooptekening te zien met aan weerskanten de berekende standen van de

optimistische buurmannen- wel zou moeten lukken met mijn computer.

Na weer enig gepraat en een kleine geïmproviseerde demonstratie begon mijn buurman enig vertrouwen te krijgen, zetten we een plan in elkaar en togen aan het werk.

Het lastigste karwei, het voorbereidende werk en het verzamelen van de benodigde gegevens, deed mijn buurman; het hameren op het toetsenbord van de QL was mijn verantwoording.

Het eerste deel van het plan omvatte het beschrijven van de 'opstallen' in onderdelen en het vermelden van eventueel nodige reparaties, onderhouds- en inspectie-termijnen. Dit eindigde in een rapport van 30 kantjes. Dit rapport kreeg de klinkende naam "Rapport Systematisch Beheer en Onderhoud."

Het rapport is volgens allerlei officiële bestekken in hoofdstukken en paragrafen verdeeld die elke een apart gedeelte van het 'gebouw' beschrijven. Hiervan zal ik alleen de hoofdgroepen aangeven:

1) Buitenriolering en drainage. 2) Bestratingen. 3) Beplantingen. 4) Terreininventaris. 5) Fundering en kruipruimtes. 6) Betonwerk. 7) Metselwerk. 8) Systeemvloeren en dakplaten. 9) Hout- en draagkonstrukties. 10) Schoorstenen en kanalen. 11) Kozijnen, ramen en deuren. 12) Gevelbekledingen. 13) Trappen en hekken. 14) Dakbedekkingen. 15) Goten en hemelwaterafvoeren. 16) Natuursteen. 17) Stukadoorswerk. 18) Tegelwerk. 19) Dekvloeren. 20) Metaalwerk. 21) Plafond- en wandsystemen. 22) Binnentimmerwerk en inrichting. 23) Beglazing. 24) Schilderwerk. 25) Binnenriolering. 26) Waterinstallaties en sanitair. 27) Verwarmingsinstallatie. 28) Elektrische installatie. 29) Klachten onderhoud.

Elk van deze zaken werd weer onderverdeeld in verschillende elementen, die allemaal bekeken werden op onderhoudstoestand en eventuele gebreken. Indien u dit zelf wilt doen is het dus zaak het gehele 'pand' in detail te bekijken en beschrijven. Elk onderdeel kreeg een eigen regel, cq 'row' in ABACUS. Het totaal kwam op zo'n 80 regels; zo'n 80 aparte te beoordelen items. Van elk item werden de volgende criteria vastgesteld:

>de urgentie van eventueel onderhoud of reparatie,
>uit welk fonds de kosten betaald moesten worden,
>of het al dan niet zelf uit te voeren was,
>of het preventief dan wel normaal onderhoud betrof, en
>of het door de bewoners of de vereniging betaald moest worden.
Bovendien werd in het rapport en in het spreadsheet steeds de cyclus van een bepaalde werkzaamheid aangegeven evenals het begin- en eindjaar daarvan.

In eerste instantie konden deze gegevens niet allemaal meer tegelijk in één spreadsheet, want hoewel het aantal cellen ruim voldoende was, alles bij elkaar was rij 1 tot 85 en kolom A t.e.m. AG nodig, was er niet voldoende geheugen om ook de 'vulling' van de cellen te bevatten. Na enige tijd gewerkt te hebben met vier stukken, waarbij de totalen van één deel steeds werden overgebracht naar het volgende deel, kwam de geheugen-uitbreiding; nu is er nog 316K over....

De eerste versie van het model moest nog gedeeltelijk met de hand ingevuld worden, als het ging om het uitrekenen van de kosten per jaar. Aan de hand van de opgegeven cyclus, begin- en eindjaar werd dan in elke jaarkolom al dan niet een bedrag ingevuld.

Eerlijk gezegd heeft het me destijds een hele zondagochtend gekost om ook daar een 'formuletje' voor te vinden. Het geheel werkt nu naar tevredenheid en berekent elke verandering geheel automatisch door in de hele planning. Het rekenen kost daardoor soms nogal wat tijd, maar zonder computer waren we er überhaupt nooit aan begonnen.

Nu hebben we een redelijk overzicht van de te verwachten kosten en kunnen we een 'spaarpotje' aanleggen door precies aangepaste maandbedragen te betalen; niet te veel en niet te weinig...
..... En dat wil toch iedereen?

De ABACUS-layout....

Kolom

- A: Urgentie: 1=direct, 2=te verschuiven.
B: Fonds: 3=onderhoudsfonds.
4=reservering.
5=geschatte post.
C: Uitvoerder: 6=laten doen.
7=zelf doen.
D: Soort: 8=normaal onderhoud.
9=preventief onderhoud.
E: Vakgroep: 1=bewoners.
2=derden.
F: Hoofdstuk nummer uit de rapportage (zie boven; no 1 t.e.m. 29)
G: Paragraaf nummer. (afhankelijk van situatie)
H: Beginjaar cyclus.
I: Duur van de cyclus in jaren.
J: Eindjaar van de cyclus.
K: R= reservering.
L: Verkorte omschrijving onderdeel.
M: Hoeveelheid (getalwaarde voor kolom N).
N: Eenheid van kolom M in m, m², st., enz.
O: Eenheidsprijs van het materiaal.

- P: Totaalbedrag materiaal.
Formule: $M\# \cdot O\#$. #=nummer van de desbetreffende rij....
- Q: Eenheid arbeids uren. (aantal uren benodigde arbeid per eenheid)
- R: Totaal uren.
Formule: $M\# \cdot Q\#$.
- S: Totaal ex BTW.
Formule: $P\# + (Q\#1 \cdot R\#)$
(\$Q1 bevat hoogte uurloon; rond de f50,-)
- T: Totaal Inkl BTW.
Formule: $\text{Int}(S\# + (T1/100 \cdot S\#)) + 0.5$
(\$T1 bevat percentage BTW)

En dan de lastigste kolommen; U t.e.m. AG. Boven de kolommen (rij 1!) staan steeds de laatste twee cijfers van het jaartal: 87 t.e.m. 99. Met kopiëren moeten alle cellen gevuld worden met een formule die aan de hand van de kolommen H, I en J al dan niet het bedrag van kolom T invult...

Voor kolom U is dit de volgende formule:

If (\$U1>=H# and \$U1<=J# and \$U1=H#+(Int((\$U1-H#)/I#)*I#),T#,0)

(de waarde van # verandert natuurlijk met elke rij; de letter U is voor elke volgende kolom anders.....)

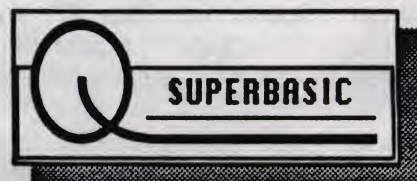
in woorden:

als het jaartal boven deze kolom groter of gelijk is aan het beginjaar (in kolom H), EN
als het jaartal boven deze kolom kleiner of gelijk is aan het eindjaar (in kolom J), EN
als het jaartal boven deze kolom gelijk is aan:
het beginjaar plus de integer van ((het jaartal boven deze kolom min het beginjaar) gedeeld door de cyclus (in kolom I)) vermenigvuldigd met de cyclus, DAN mag het bedrag inclusief BTW ingevuld worden (uit kolom T) en ANDERS niets (0).

Onderaan wordt natuurlijk elke jaarkolom opgeteld zodat de totale kosten per jaar direkt af te lezen zijn. (SUM(U#:U##).

Ik ben mij er van bewust, dat het niet eenvoudig zal zijn om een rapportage zelf te maken; daarvoor is enig verstand van bouwkundige zaken nodig. Het invullen en vullen van ABACUS zal niet zo'n probleem zijn. Gaat u daadwerkelijk aan de slag (in de slag) met dit model en deze toepassing, dan ben ik altijd bereid om een lege, geformatteerde cartridge te vullen met zowel de schriftelijke rapportage als de gegevens uit onze specifieke situatie; misschien komt u daar een eind verder

mee. Worden de vragen, op bouwkundig vlak, moeilijker, dan kan ik misschien nog eens mijn ook dan hopelijk optimistische buurman om enkele richtlijnen vragen.....



Drie Functies voor omreken problemen.

De cursus machinecode gaf u vorige keer een aantal omreken problemen. In de volgende drie functies van Hans Brinkmann heeft u handige hulpjes voor het omrekenen naar het sexagesimale stelsel. (heet dat zo?)

De functie DEC_HMS zorgt ervoor dat een getal wordt omgezet in een getal, uitgedrukt in uren (of graden), minuten en seconden.
Bijv. 13.5125 wordt 13.3045 (13 u/g , 30 min , 45 sec).

De functie HMS_DEC doet het omgekeerde, de functie SGN spreekt voor zich.

Ik ben trouwens benieuwd of iemand een nog kortere versie van SGN kan maken.

Veel QL-plezier gewenst uit Den-haag.
HANS BRINKMANN

```
10 DEFine FuNction SGN (r)
20 RETurn ((r<0)-(r<0)*2)
30 END DEFine
```

```
10 DEFine FuNction DEC_HMS (g)
20 pn0=SGN(g) : g_pos=ABS(g) : g_int=INT(g_pos)
30 g_rest=g_pos-g_int:g_rest=INT(g_rest*3600+.5)
40 min=INT(g_rest/60) : sec=g_rest-(min*60)
50 IF sec=60 : min=min+1 : sec=0 : END IF
60 IF min=60 : g_int=g_int+1 : min=0 : END IF
70 RETurn (g_int+(min/100)+(sec/10000))*pn0
80 END DEFine
```

```
10 DEFine FuNction HMS_DEC (g)
20 pn0=SGN (g) : g_pos=ABS(g) : g_int=INT(g_pos)
30 g_rest=(g_pos - g_int)*10000
40 m_dec=INT(g_rest/100)
50 s_dec=g_rest-(m_dec*100)
60 RETurn (g_int+(m_dec*60+s_dec)/3600)*pn0
70 END DEFine
```


- 3 t/m 12 Naam van de cartridge: 10 karakters
evt. opgevuld met spaties
- 13 t/m 14 16 bit random (willekeurig) nummer

De identifikatiecode is voor QDOS belangrijk om te kunnen herkennen dat het inderdaad om een sectorheader gaat. Het sectornummer is vanzelfsprekend nodig om de verschillende sectoren uit elkaar te kunnen houden. Het random nummer wordt door QDOS bij het formatteren willekeurig gekozen en is bedoeld om twee bandjes met dezelfde naam toch uit elkaar te kunnen houden. Het nummer wordt ondermeer gebruikt om te zien of het bandje is verwisseld. Als je namelijk de eerste keer een DIR geeft wordt een gedeelte van het RAM geheugen gereserveerd voor gegevens over de betreffende cartridge en een deel van de inhoud van de microdrive wordt gekopieerd naar dat geheugen. Als je dan de tweede keer een DIR geeft hoeft de QL alleen maar te zien of er nog steeds hetzelfde bandje inzit door naar dat nummer te kijken (staat op elke sector en is dus snel gevonden) en te vergelijken met de gegevens in het geheugen.

De eigenlijke informatie wordt dan uit het geheugen gehaald in plaats van het opnieuw van cartridge te lezen. Die tweede DIR gaat dus veel sneller.

3) Het datagebied

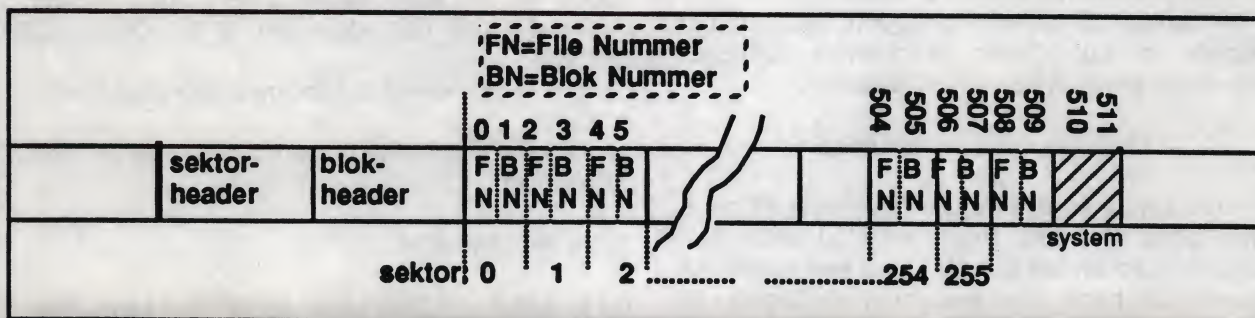
Het datagebied bestaat uit 512 posities. Hoe deze 512 posities gebruikt worden hangt af van het type sector. De sector kan namelijk gebruikt worden als 'mapping sector' (plattegrond), als 'directory sector' (inhoudsopgave) en als 'datasector' (gegevens). Wat dit betekent en hoe dat gaat zullen we verderop bekijken.

3A Datasectors

Allereerst moeten we even weten hoe een programma of bestand (vanaf nu file genaamd) opgeslagen wordt. Elke file krijgt bij opslag een nummer. De eerste file die je opbergt krijgt nummer 1, de volgende nummer 2 enz. Echter, als je file nummer 1 uitveegt en er een derde file bijzet, dan krijgt die weer nummer 1.

Bij een DIR commando komen de files ook in de volgorde van het filenummer en dus niet altijd in de volgorde waarin je ze op de band gezet hebt. File nummer 0 bestaat altijd en is een speciale file, de zogenaamde directory file; hier komen we later op terug.

Zoals al gezegd knipt de QL de file in stukken van 512 karakters en schrijft zo'n stuk (een blok) weg op een sector. Het laatste blok hoeft niet compleet



Figuur 2: Opbouw van een mapping-sektor.

2) De blokheader.

De blokheader bestaat uit slechts twee karakters, nl het file (programma) nummer en het bloknummer.

De QL knipt namelijk een bestand of programma in stukken (blokken) om het op te kunnen slaan. Op een sector moet dus staan van welke file het hoeveelste blok er op staat. De blokheader wordt bij elke schrijfofdracht naar de betreffende sector bijgewerkt.

Positie Betekenis

- 1 Programma (file) nummer (0-255)
- 2 Bloknummer (0-255)

te zijn. Elk blok krijgt ook een nummer, het eerste is altijd blok 0. Dus als file nummer 10 bestaat uit 3 blokken dan zijn er 3 sectoren te vinden met in de blokheader (zie fig1) <10, 0> <10, 1> en <10, 2>.

Belangrijk is dat als een file wordt weggeschreven er ook een fileheader (kop) wordt gemaakt. Deze fileheader is altijd 64 karakters lang en bevat gecodeerde informatie over de file. Dat betekent dat in het eerste blok (blok 0) van de file niet 512 maar $512 - 64 = 448$ karakters over zijn voor de echte inhoud van de file. Alle volgende blokken (1, 2 etc) kunnen wel 512 karakters bevatten met de inhoud van de file.

OVER MICRODRIVES EN SECTOREN.

De cartridge; geprezen en verguisd. Jan en Kees leren u de diepste geheimen van dit wonderdje met wieljes..... en vertellen gaandeweg hoe de QL er gebruik van maakt.

Dit stukje is een bewerking van het artikel van Colin Ople uit het Januari nummer van QL USER uit 1986. Dit laatste nummer voor de overgang naar QL WORLD heeft niet iedereen bereikt en is toch erg interessant voor diegenen die iets meer willen weten over de manier van opslaan op microdrive. Met de machinecodeprocedures uit het artikel heb ik toch wel leuk kunnen spelen, vandaar deze reactie.

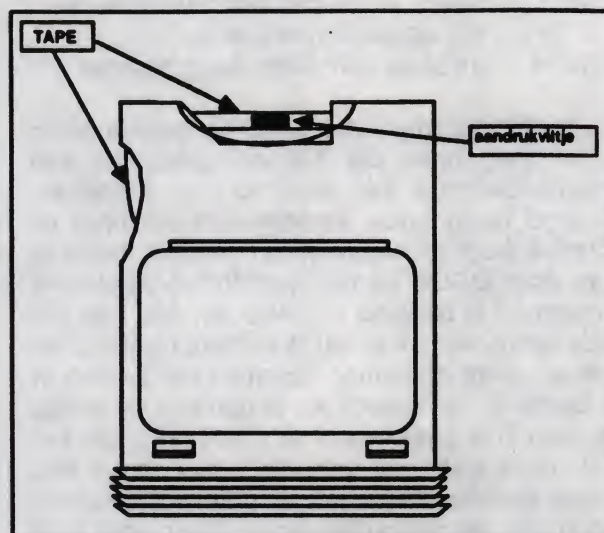
De bijbehorende Basicprogramma's waren in eerste instantie bedoeld om een per ongeluk verwijderd (delete) bestand of programma weer mee tevoorschijn halen of in geval van een slechte sector (BAD OR CHANGED MEDIUM) nog wat te redden.

Volgens ons werken ze lang niet altijd goed, en we werken nu aan een betere versie. In dit verhaal komt daarom eerst aan de orde hoe microdrives georganiseerd zijn en aan het eind van deze aflevering nog een stukje over het gebruik van de machinecode routines. Je kunt dan tenminste zelf ook aan het spelen en experimenteren.

In een volgende aflevering hopen we dan het complete in SuperBasic geschreven cartridge-dokter programma te kunnen presenteren.

Opbouw van een cartridge

In het bekende zwarte doosje (cartridge) zit ca. 5 meter band verwerkt. Begin en eind zitten aan elkaar geplakt en het geheel zit op een enkele rol opgerold. De band wordt vanuit het midden van de spoel via een wat rare kronkel naar buiten getrokken, langs de kop geleid, gaat de hoek om langs de aandrijfrol, en komt dan op de buitenkant van de rol weer terecht. Omdat de band uit de kern wordt weggetrokken gaat het geheel draaien, zodat de op de buitenkant terugkomende band ook weer opgerold wordt. Als je eens een defekte cartridge hebt, sloop hem dan maar eens open en verbaas je erover dat dat allemaal goed gaat bij een bandsnelheid van 70 centimeter per seconde.



Opbouw van een sector

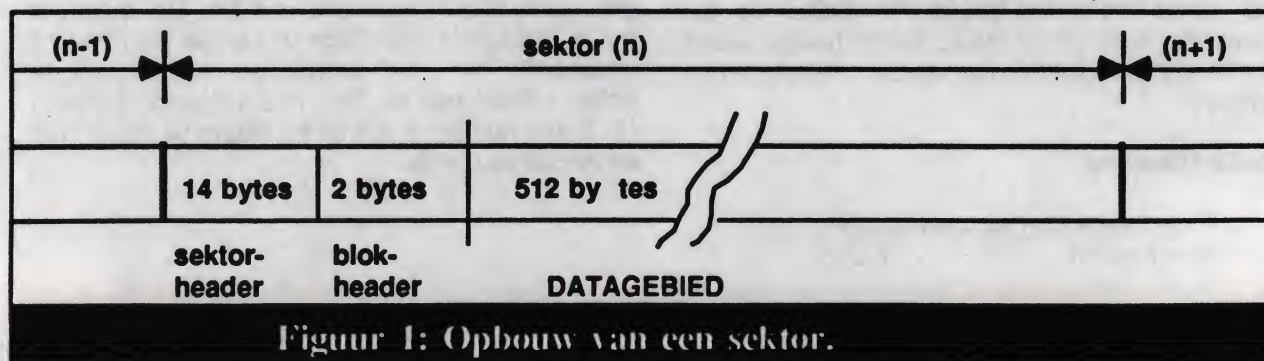
Zie figuur 1. Als je een cartridge formatteert wordt de cartridge ingedeeld in sectoren waarbij elke sector een nummer krijgt. Vanwege de onzekerheden in de exacte lengte van de band en de snelheid van de microdrive, begint het nummeren van de sectoren bij 255 en loopt af tot 0. Zoveel sectoren kunnen er beslist niet op, dus wanneer de QL bij ongeveer 30 is schrijft hij gewoon over de eerste (hoge) nummers heen. De laatste is dan altijd sector 0. Het totaal aantal sectoren loopt dan dus van 0 tot circa 220 sectoren.

Op een sector kun je in principe 3 gebieden onderscheiden: een sectorheader (kop), een blokheader en een datagebied.

1) De sectorheader.

De sectorheader bestaat uit de eerste 14 karakters en wordt eenmalig beschreven, namelijk tijdens het formatteren. Daarna kan hij alleen nog gelezen worden (met speciale machinecode routines). De indeling van deze 14 karakters is als volgt:

Positie	Betekenis
1	Identificatiecode (Decimaal 255, Hexadecimaal \$FF)
2	Sectornummer: 0 tot max 255 (meestal minder, ca 220)



Figuur 1: Opbouw van een sector.

3C De DIRECTORY file

Op elk bandje staat een directoryfile. Zie fig 3. Deze file wordt direkt na het formatteren aangemaakt en heeft het nummer 0. Deze wordt door de QL gebruikt om bij te houden welke files er op het bandje staan. Het is immers handiger om ergens een inhoudsopgave te hebben staan; anders moet je elke keer de hele band afzoeken naar het begin van de file die gezocht wordt. Bovendien is het voor de buitenwereld (wij dus) veel plezieriger om files een naam te geven, terwijl de QL liever met een nummertje werkt. Ergens moet er een verband gelegd worden tussen die twee dingen, en ook dat kan in zo'n inhoudsopgave geregeld worden. Deze file is doorgaans niet toegankelijk voor gebruikers (wel met bijv. het Toolkit2 commando OPEN_DIR).

Voor elke file is er ruimte voor 64byte aan informatie. Jawel, dezelfde 64 karakters als in de fileheader. Dit betekent dat van elke file de file-informatie twee keer op het bandje staat nl in blok 0 van de file (fileheader) en in de directoryfile (dus op een andere sector). Dit geldt niet voor file 0, want file 0 IS de directory file en voor deze directory file is de fileheader en de directory een en hetzelfde gebied.

Om de fileinformatie te pakken moet je dus het file-nummer weten en in de directoryfile op positie $64 \cdot \text{filenumber} + 1$ beginnen met lezen. Als de directory file langer dan een sector moet worden (omdat er meer files worden aangemaakt), wordt er een tweede sector gereserveerd (file 0 blok 1) enz.

Tot zover het verhaal over de organisatie van de verschillende soorten data op een cartridge. Nu wordt het tijd om die zaken aan elkaar te knopen en te bekijken wat er gebeurt als je een file schrijft, leest of weer uitpoetst.

Schrijven van een file.

Wat is nu de gang van zaken als je iets naar micro-drive schrijft?

Een deel van de fileheader kan al aangemaakt worden; de naam van de file heb je zelf opgegeven en QDOS kijkt welk filenummer er toegekend moet worden: het laagste nummer dat niet (meer) gebruikt wordt. Een deel van de fileheader kan dus al ingevuld worden op de eerstgevonden vrije (en bruikbare) sector en ook de eerste 448 bytes kunnen daar nog bijgestopt worden. De rest van de bytes worden opgeslagen op andere sectoren. Als je 't later bekijkt dan zal blijken dat QDOS voor het opslaan van de rest van de informatie niet de volgende sector (dwz met een sectornummer dat 1 lager ligt) maar er 12 overslaat (dus een sectornummer dat 13 lager ligt). Zie het stukje over het lezen van een file. Als alle informatie op sectoren is weggeschreven dan wordt ook de lengte van de file (inklusief de 64 byte van de header) ingeschreven. Desectormap kan nu ook bijgewerkt worden. Tenslotte wordt van elke nieuw be-

schreven sector nog gecontroleerd of het inderdaad goed is overgekomen. Als deze verifikatie niet klopt dan krijg je de boodschap "bad or changed medium". Als je alleen vergeten was dat je het "write protect" lipje had afgebroken dat kan die fout eenvoudig hersteld worden. Als dat niet het geval was dan wordt het tijd om de inhoud van de cartridge even naar elders te kopiëren en de weerspannige cartrdige opnieuw te formatteren. Het hele bandje wordt dan weer op kwaliteit gecontroleerd.

Lezen van een file.

Het lezen van een file gaat net omgekeerd in z'n werk. Je geeft de naam van de file op en die wordt in de directoryfile opgezocht. De directoryfile kan gevonden worden door in de sectormap te zoeken naar de file met nummer 248, block 0. Daar staat ook de lengte van de directoryfile in zodat bekend is tot hoever er doorgezocht moet worden naar eventuele volgende blokken. Als de naam gevonden is, is ook het filenummer af te leiden. In de mapping sector kan nu op zoek gegaan worden naar het groepje met het juiste filenummer en bloknummer 0. Die sector kan dan worden binnengehaald (behalve de eerste 64 bytes) enz. Uit de lengte van de file kan worden afgeleid tot welk bloknummer doorgezocht moet worden. Wanneer meerdere blokken na elkaar moeten worden gelezen dan blijkt dat het handig is dat ze niet allemaal vlak achter elkaar staan. Bij het inlezen van een SuperBasic programma duurt het vertalen van ASCII naar interne codes namelijk zo lang dat de band beter even stilgezet kan worden. De tussenruimte van 12 sectors is nu zo gekozen dat de microdrive even gestopt kan worden, weer gestart en dat de band weer op volle snelheid is als het eerstvolgende blok met info langskomt.

Elke sector bevat naast de informatiebytes ook nog een twee-byte checksum. Als de checksum die op de sector staat niet klopt met het getal dat QDOS uit de zojuist gelezen 512 informatiebytes heeft afgeleid, dan wordt gekonkludeerd dat er iets mis is. De lees poging van die sector wordt dan nog een aantal keer herhaald en als het nog steeds niet klopt dan wordt gestopt met "bad or changed medium". Dan zit je met een probleem; je mist een informatieblok. Over het algemeen stopt QDOS dan met lezen zodat niet alleen die ene sector maar ook eventuele volgende blokken worden niet meer gelezen worden. Alleen als een file met het LBYTES kommando gelezen wordt dan leest QDOS niet strikt op volgorde van bloknummer en ben je alleen de slechte blokken kwijt.

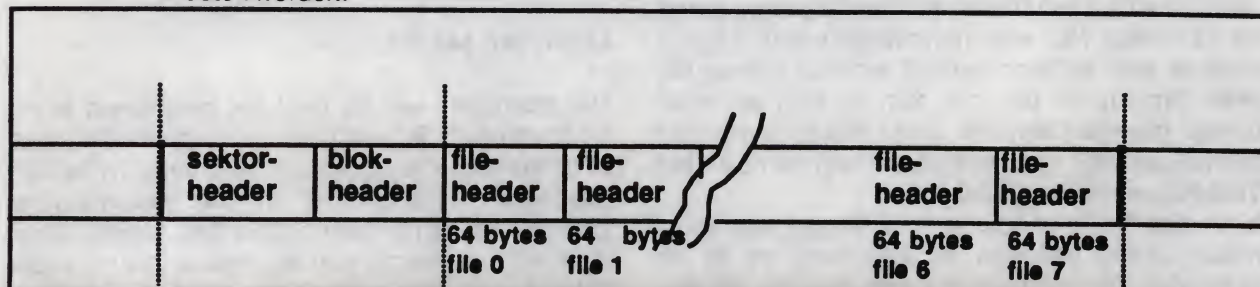


De opbouw van de fileheader is als volgt:

Positie	Betekenis
1 t/m 4	32 bit getal dat de lengte van de file aangeeft (inclusief de file header)
5	File toegangscode: (normaal 0); geen idee waarvoor dat is
6	File typecode: 0 voor BASIC programma of datafiles, 1 voor machinetaal programma's die met EXEC gestart moeten worden.

de cartridge kunnen, worden de laatste twee karakters gebruikt door de QL; waarvoor weet ik niet.

Zoals al gezegd krijgt elke file een nummer, maar op de plaats van het filenummer in de mapping sector kunnen ook wat speciale filenummers staan; zie onderstaand lijstje.



Figuur 3: Opbouw van een directory-sector.

7 t/m 14	8 Karakters voor file informatie. Als het type 1 is (executable), dan staat op positie 7 t/m 10 de hoeveelheid geheugen die gereserveerd moet worden.
15 t/m 16	2 bytes (1 word) voor de lengte van de filenaam.
17 t/m 52	Maximaal 36 karakters voor de naam van de file.
53 t/m 56	Datum en tijd dat de file wordt geschreven (in seconden vanaf 1961 1 januari 00:00:00 uur. In BASIC om te zetten naar datum met DATE\$(getal)). Op een QL zonder Toolkit wordt dit niet gebruikt.
57 t/m 64	Gereserveerd gebied van 8 karakters. Het was ooit de bedoeling om hier nog meer info over de tijd te plaatsen.

Filenummer Betekenis dec hex

0	\$00 directory file
248	\$F8 mapping file; altijd sector 0, altijd 1 blok lang.
253	\$FD sector wordt niet (meer) gebruikt.
254	\$FE sector is slecht.
255	\$FF sector bestaat niet (je krijgt immers maar circa 220 sectors op een bandje)

Zoals hierboven al aangegeven wordt bij het formatteren nagegaan hoeveel sectoren er op het bandje geschreven kunnen worden. De niet-bestaande sectoren worden in de mappingsector ook als 'niet bestaand' geboekt. Het formatter-proces schrijft niet alleen de sectorheaders maar controleert ook of het tussenliggende gebied (waar later de data moet worden geschreven) wel okay is. Er wordt dan een testpatroon geschreven, en als dat niet goed teruggelezen kan worden dan wordt die sector als 'bad' (slecht) aangemerkt. Die informatie wordt in de mapping sector opgeslagen door op de met die sector overeenkomende plaats filenummer 254 (bloknummer 0) te schrijven. QDOS weet dan later dat die sector niet gebruikt moet worden.

3B De MAPPING sector

Er is altijd een sector met nummer 0. Deze sector wordt op een speciale manier ingedeeld en gebruikt en heet de 'mapping' sector. Zie figuur 2. In deze sector die normaal niet toegankelijk is voor gebruikers houdt de QL bij op welke sector welk deel van een file staat. Een soort routebeschrijving dus. De inhoud van deze sector bestaat uit 256 groepjes van steeds twee bytes, waarbij het eerste byte het filenummer en het tweede byte het bloknummer voorstelt. De plaats van het groepje in de mappingsector geeft aan op welk sectornummer die informatie betrekking heeft.

Groepje 0 slaat op sector 0, groepje 1 op sector 1 etc. Dus als van filenummer 10 het bloknummer 3 staat op sector 8, dan staat in de mapping sector (sector 0) in groepje 8 het filenummer (10) en het bloknummer (3). Groepje 8 bevindt zich op positie 17 en 18 van de mapping sector (als we bij positie 1 beginnen te tellen). Omdat er nooit 255 files op

Na het formatteren krijg je een mededeling zoals '218/221 sectors', waarmee aangegeven is dat er in totaal 221 sectors gemaakt zijn en dat er daarvan 218 goed zijn bevonden. Bij een DIR commando op een zojuist geformatteerd bandje krijg je altijd twee sectoren minder beschikbaar dan aanwezig; in het voorbeeld dus 216/218. Dit kan nu verklaard worden; er zijn immers twee speciale sectoren aangemaakt: de mapping sector (file 248 op sector 0) en de directory file (file nummer 0) die ook minimaal een sector lang is.

MACHINECODE

```
10110101101000101010001010
10101010000111101010011001
10110001100111110100110101
01011110001111000110001101
10101010101011111000111000
```

HOE DOE JE HET IN ASSEMBLY ?

Inhakend op het soort spreuk dat we elke dag op de achterruit van de auto voor ons kunnen zien probeert Marc u aan een originele slagzin te helpen.

Afgelopen maand heb ik maar een paar telefoontjes gehad van mensen, die een vraag stelden over assembly. Het geringe aantal heeft me aangezet om dit artikel te schrijven. (Tussen haakjes: ik zit er niet op te wachten dat mijn telefoon roodgloeiend staat.)

Bij dit artikel staan twee listings.

Listing 1 is een kleine library voor eenvoudige I/O.

Listing 2 is veel belangrijker; hierop staat een klein programma dat gebruik maakt van de I/O-library. Het programma is bedoeld om te laten zien dat veel operaties lang niet zo moeilijk te realiseren zijn (zeker niet als je gebruik maakt van een library).

Ik hoop dat ik hiermee de drempel kan verlagen om iets in assembly te schrijven.

Met de twee listings wordt gelijk een voorbeeld van separaat assembleren gegeven. Mocht u onderstaand verhaal te moeilijk vinden, dan kunt het volgende doen:

- * tik de beide listings in 1 file
- * verwijder alle regels waarin 'xdef' en 'xref' voorkomt
- * gewoon assembleren

Het idee achter het separaat assembleren is, dat van iedere file die assembly-code bevat een "link file" wordt gemaakt. Deze file bevat gewone machine-code en eventuele verwijzingen naar locaties in andere files.

Een linker maakt van alle link-files die bij elkaar horen op zijn beurt een file met pure machine-code.

De assembler kan niet zomaar opmerken dat u een verwijzing maakt naar een locatie in een andere file (normaliter geeft de assembler een foutmelding als hij een label niet kan vinden).

Om de assembler duidelijk te maken dat u gebruik wilt maken van 'iets' dat in een andere file staat

bestaat de 'xref directive'.

De syntax is: **xref <extern label>**

Het keyword **XREF** staat voor **eXternal REFerence**.

Het is ook mogelijk om tegen de assembler te zeggen: "ik heb een label waar een ander gebruik van mag maken". Met andere woorden:

"ik heb een subroutine (of locatie) waarnaar een subroutine in een andere file naar verwijst.

De syntax is: **xdef <label>**

Het keyword **XDEF** staat voor **eXternal DEFinition**.

Nu weer een voorbeeld uit de praktijk: ik weet dat er een routine in de library staat, die een window opent en het window-id teruggeeft in register A0.

Deze routine heet 'open_scr'. Mijn programma ziet er dan als volgt uit:

```
xref open_scr ...
bsr open_scr move.l a0, ...
```

In de source-file van de library staat dan de volgende code:

```
xdef      open_scr ...
open_scr  moveq #io_open,d0
...        trap #2
           rts      ; id in a0
```

Ik ben niet goed op de hoogte in hoeverre de verschillende assemblers link-files kunnen produceren. Ik werk met een assembler en linker van Computer One en ben hier zeer tevreden over.

Marc Kool

LISTING 1

```
;
;
; library module to use with Computer One (and
;                                     hopefully other)
;                                     assemblers
;
; (C) by Marc H. Kool april 1987, copies may be
;                                     distributed freely
;
; this module contains the following subroutines:
;
; wnd_open  open a default window
; con_open  open a default console window
; pr_char   print a character
; pr_enter  print an enter (go to next line)
; pr_int    print an integer value in characters
; pr_str    print a string
; get_char  read a character
; kill_me   kill this job
```


Verwijderen van een file.

Wat doet nu de QL bij het verwijderen van een file?
Twee dingen:

a) De eerste 16 bytes van de fileinformatie in de DIRECTORYfile worden gevuld met nullen. Dit heeft als gevolg dat de lengte van de filenaam niet meer bekend is (positie 15 en 16) maar dat de naam van de file zelf nog wel in de directory staat (vanaf positie 17 en verder). Het is nu alleen zo dat een DIR commando de file niet meer 'ziet'.

b) In de mapping sector worden in de corresponderende groepjes van twee bytes de filenummers en bloknummers gewijzigd in \$FD00 dwz filenummer 253 (lege sector) en bloknummer 0. WEL aanwezig zijn dus nog de naam van de file in de directory file, de daadwerkelijke inhoud van de file (inklusief de volledige fileheader) in de betreffende sectoren en de blokheaders van die sectoren.

Vandaar dat het uitvegen van een file ook veel sneller is dan het schrijven ervan. Als er na het uitvegen weer een programma op het bandje wordt gezet dan loop je de kans dat sectoren van de uitgeveegde file worden overschreven met blokken van de nieuwe file. Dat moet je dus niet doen als je de uitgeveegde file weer terug wilt toveren.

Herstellen van een uitgeveegde file.

Wat te doen bij "bad or changed medium".

In het bovenstaande is alle informatie om problemen als "bad or changed medium" of een per ongeluk weggegooid file aan te pakken. Het door Colin Opie geschreven programma is echter in veel gevallen niet afdoende en doet het bovendien niet erg handig. Vandaar dat er nog even wat aan gesleuteld moet worden en in de volgende aflevering hopen we een volledige "cartridge dokter" te kunnen presenteren.

Gebruik van GET_MSEC en PUT_MSEC.

Allereerst moeten de SuperBasic uitbreiding ingelinkt worden. Dat gaat met:

```
adres=RESPR(900)LBYTES
mdv1_getsec2_code,adresCALL adres
```

Je hebt nu 2 nieuwe procedures tot je beschikking, GET_MSEC om een sector te lezen en PUT_MSEC om een sector te schrijven.

De procedures hebben 3 parameters:

- het nummer van de microdrive. 1 voor MDV1 en 2 voor MDV2;
- het nummer van de sector die gelezen/geschreven moet worden (0 t/m het hoogste bestaande sectornummer);
- wat er gelezen of geschreven moet worden (in stringformaat van 514 karakters).

Voorbeeld:

Voor het lezen van sector nummer 13 van MDV1_ in sector\$ gebruik je:

```
DIM sector$(514)GET_MSEC 1,13,sector$
```

Nu staat de inhoud van de sector in sector\$(1) t/m sector\$(512). Het bloknummer staat in sector\$(513) en het filenummer in sector\$(514).

Het file- en bloknummer krijg je in decimaal formaat met:

```
bloknummer=CODE(sector$(513))
filenummer=CODE(sector$(514))
```

Als QDOS er niet in geslaagd is de sector terug te lezen dan staat er <TE> in sector\$(1 TO 4); een afkorting van 'transmit error'.

Voordat je sector\$ gaat gebruiken is het raadzaam om eerst op <TE> te testen, anders lees je onzin. Voor het schrijven van een sector zul je eerst een string moeten vullen met de benodigde informatie:

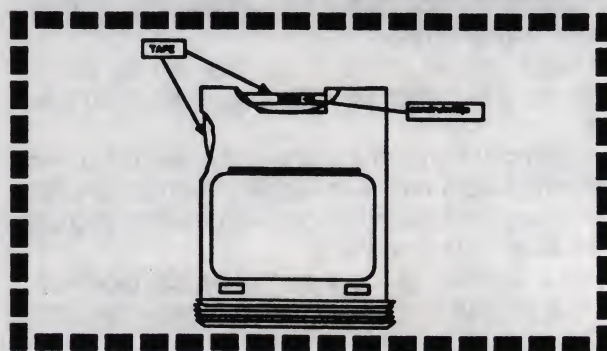
```
sector$(1 TO 512) met dechte informatie;
sector$(513) met het bloknummer en
sector$(514) met het filenummer.
```

Dat kan vervolgens als filenummer 3 blok 2 op sector 13 van MDV1_ worden weggeschreven met:

```
DIM sector$(514)sector$(1 TO 512)=FILL$
('?',512)
filenummer=3
bloknummer=2
sector$(513)=CHR$(bloknummer)sector$(514)=CHR$(filenummer)
PUT_MSEC 1,13,sector$
```

Wees voorzichtig met PUT_MSEC! Je kunt QDOS er flink mee in de war helpen! Om het experimenteren wat te vergemakkelijken is er het programma MDVINSPECT_bas. Het leest de mapping sector en aan de hand van een menuutje kun je de sectoren verder bekijken. De programma's zijn via de cartridgeservice te bestellen.

Jan Kortschot, Kees vander Wal



; the following routines are supposed to be in some
other module and
; must be linked together with this program by a
suitable linker

```
xref  con_open, pr_str, pr_char, pr_enter,
      pr_int
xref  get_char, kill_me
```

; the following constants are offsets for my variables
relative to a5

```
con_id equ 0
char_count equ 4
character equ 6
```

```
lea  vars,a5      ; all my variables
                  ; are relative to a5
```

```
bsr  con_open     ; open a console
                  ; window
bne  exit         ; if error then stop
move.l a0,con_id(a5) ; store the channel-
                  ; id
```

```
lea  message,a1   ; print a welcome
                  ; message
```

```
bsr  pr_str
```

```
loop  clr.w  char_count(a5) ; char_count = 0
      ; repeat
      move.l con_id(a5),a0 ; read a character
      bsr  get_char
      move.b d1,character(a5) ; store it at a safe
                              ; place
      addq.w #1,char_count(a5) ; increment
                              ; character counter
      bsr  pr_char         ; print the character
                              ; the channel-id
                              ; was still in a0
      cmp.b #'!',character(a5) ; if it was no '!'
                              ; then repeat loop
      bne.s loop          ; until character =
                              ; '!'
```

```
bsr  pr_enter     ; start at a new line
move.w char_count(a5),d1 ; print the number
                        ; of characters
bsr  pr_int       ; that I have read
```

```
exit  bra  kill_me ; stop this program
```

```
vars  ds.b  8      ; reserve space for
                  ; my variables
```

```
ENTER equ 10      ; character constant
```

```
message dc.w 51
        dc.b "Type een paar karakters. Stop met
              een
              uitroepteken",ENTER
```

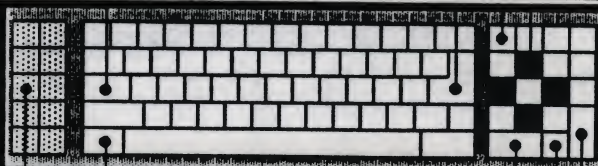
even



Schön, oder?

Betaalbare diskdrives...

Wim Spanjer en Fred Brands zien kans om bij voldoende interesse originele NEC-drives te leveren tegen lage prijzen. Het gaat dan om **TWEE 3,5 inch drives inclusief voeding, kast en bandkabel**. Bij minimaal VIJF geïnteresseerden komen de kosten op **f675,-**. (Een bijbehorende disk-controller zou u bij Spanjer op 275,- komen). Neem even contact op met Willem Spanjer... **03404 - 20581**.



Betaalbare toetsenborden...

Eveneens bij voldoende interesse kan Willem Spanjer voor een **toetsenbord a-la-IBM-XT** zorgen (inclusief de dan benodigde print) voor **f250,-**. Deze prijs kan gehaald worden als er minimaal 25 leden zich aanmelden en garant willen staan voor afname van zo'n toetsenbord. Alweer: bel **03404 - 20581**.

; the following definitions make it possible to link these subroutines

```
xdef wnd_open, con_open
xdef pr_char, pr_enter, pr_int, pr_str
xdef get_char, kill_me
```

; QDOS definitions:

```
ut_con equ $C6 ; open console
ut_scr equ $C8 ; open screen
ut_mint equ $CE ; print integer

mt_frjob equ $05 ; kill job

io_fbyte equ $01 ; read byte
io_sbyte equ $05 ; write byte
io_sstrg equ $07 ; write string
sd_cure equ $0E ; turn cursor on

ENTER equ 10 ; character constant
```

```
wnd_open lea par_block,a1 ; open a default
; window

move.w ut_scr,a2
jsr (a2) ; window-id in a0 (if
; succesfull)

tst.l d0 ; set condition codes
; to check for

rts ; an error and return to caller
```

```
con_open lea par_block,a1 ; open a default
; console window

move.w ut_con,a2
jsr (a2)
tst.l d0 ; check for an
; error...

bne.s error
moveq #sd_cure,d0 ; turn the cursor on
moveq #-1,d3
trap #3
tst.l d0 ; set condition code

error rts
```

```
pr_char moveq #io_sbyte,d0 ; print the character-
; value in d1

moveq #-1,d3 ; channel-id must be
; in a0

trap #3
tst.l d0 ; set condition code
; to check error

rts
```

```
pr_enter moveq #ENTER,d1 ; go to next line
bsr.s pr_char ; channel-id must be
; in a0

rts
```

```
pr_int move.w ut_mint,a2 ; print an integer
; value in d1

jsr (a2) ; channel-id must be
; in a0

tst.l d0
rts
```

```
pr_str moveq #io_sstrg,d0 ; print a string
; pointed by a1

moveq #-1,d3 ; channel-id must be
; in a0

move.w (a1)+,d2 ; put length of
; string in d2

trap #3
tst.l d0 ; set condition codes

rts
```

```
get_char moveq #io_fbyte,d0 ; read a byte from a
; channel into d1

moveq #-1,d3 ; channel-id must be
; in a0

trap #3
tst.l d0 ; set condition codes

rts
```

```
par_block dc.b 5,2 ; border colour &
; width

dc.b 0,5 ; paper & ink
; colours

dc.w 480,200,20,10 ; window
; width+height &
; X,Y origin
```

```
kill_me moveq #mt_frjob,d0 ; kill this job

moveq #-1,d1
moveq #-1,d3
trap #1 ; there is no return
; from here...
```

LISTING 2

; this is an example program to show how to use a library module

; the example consists of:

```
;
; opening a console window
; print a welcome message
; repeat
; read a character
; print that character
; until the character is '!'
; print the number of characters entered
; stop the program
;
```

; made by Marc H. Kool, april 1987


```

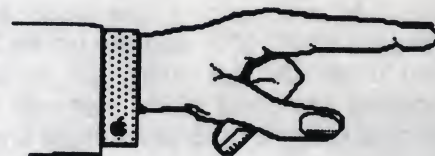
22 IF dir$(c)='---':EXIT c
23 END FOR c
24 CLOSE#7
25 c=c-1
26 CLS#2:PRINT#2,DISK: 'dir$(0);' 'dir$(1);' vrij\"
27 FOR n=1 TO c-1 STEP 2
28 IF LEN(n)=1:PRINT#2,'';
29 PRINT#2,n!dir$(n+1)::a=20-
  LEN(dir$(n+1)):PRINT#2,FILL$( 'a);
30 IF LEN(n+1)=1:PRINT#2,'';
31 PRINT#2,n+1!dir$(n+2)
32 IF dir$(n)='---':EXIT n
33 END FOR n
34 END DEFine
35 :
36 DEFine PROCedure driver1
37 OPEN#3,ser1
38 BAUD 9600
39
  PRINT#3,CHR$(27);CHR$(64);CHR$(27);CHR$(
  33);CHR$(4);CHR$(27);CHR$(83);CHR$(0);CHR
  $(27);CHR$(65);CHR$(x);
40 END DEFine
41 :
42 DEFine PROCedure driver2
43 OPEN#3,ser1
44 BAUD 9600
45
  PRINT#3,CHR$(27);CHR$(64);CHR$(27);CHR$(
  33);CHR$(4);CHR$(27);CHR$(65);CHR$(x+4);
46 END DEFine
47 :
48 DEFine PROCedure print1
49 direc
50 IF y=1:driver1
51 IF y=2:driver2
52 PRINT#3,DISK: 'dir$(0):PRINT#3,dir$(1);' vrij\"
53 FOR n=1 TO c-1
54 IF LEN(n)=1:PRINT#3,'';
55 PRINT#3,n!dir$(n+1)
56 IF dir$(n)='---':EXIT n
57 END FOR n
58 END DEFine :PRINT#3,"\\":GO TO 3
59 :
60 DEFine PROCedure print2
61 direc
62 IF y=1:driver1
63 IF y=2:driver2
64 PRINT#3,DISK: 'dir$(0);' 'dir$(1);' vrij\"
65 FOR n=1 TO c-1 STEP 2
66 IF LEN(n)=1:PRINT#3,'';
67 PRINT#3,n!dir$(n+1)::a=20-
  LEN(dir$(n+1)):PRINT#3,FILL$( 'a);
68 IF LEN(n+1)=1:PRINT#3,'';
69 PRINT#3,n+1!dir$(n+2)
70 IF dir$(n)='---':EXIT n
71 END FOR n
72 END DEFine :PRINT#3,"\\":GO TO 3
73 :
74 DEFine PROCedure print3
75 direc
76 IF y=1:driver1
77 IF y=2:driver2

```

```

78 PRINT#3,DISK: 'dir$(0);' 'dir$(1);' vrij\"
79 FOR n=1 TO c-1 STEP 3
80 IF LEN(n)=1:PRINT#3,'';
81 PRINT#3,n!dir$(n+1)::a=20-
  LEN(dir$(n+1)):PRINT#3,FILL$( 'a);
82 IF LEN(n+1)=1:PRINT#3,'';
83 PRINT#3,n+1!dir$(n+2)::a=20-
  LEN(dir$(n+2)):PRINT#3,FILL$( 'a);
84 IF LEN(n+2)=1:PRINT#3,'';
85 PRINT#3,n+2!dir$(n+3)
86 IF dir$(n)='---':EXIT n
87 END FOR n
88 END DEFine :PRINT#3,"\\":GO TO 3
89 :
90 DEFine PROCedure menu
91 CLS:PRINT' Instellen van de regelfstand,\ type 6,
  7 of 8 (.72 inch bij mini,\ 4+ .72 inch bij het
  andere type).\ SPATIE = vorige waarde\ ENTER
  =default instelling'
92 a=CODE(INKEY$(-1))
93 SElect ON a
94 =32:GO TO 101
95 =10:GO TO 2
96 =54:x=6
97 =55:x=7
98 =56:x=8
99 =REMAINDER :piep:GO TO 92
100 END SElect
101 CLS:PRINT' Instelling schrifttype:\ type 1
  voor mini, of 2 voor "groot"\ SPATIE = vorige
  waarde.'
102 a=CODE(INKEY$(-1))
103 SElect ON a
104 =32:GO TO 109
105 =49:y=1
106 =50:y=2
107 =REMAINDER :piep:GO TO 102
108 END SElect
109 CLS:PRINT' Instellen aantal kolommen in print\'
  type 1, 2 of 3\ SPATIE voor vorige waarde.'
110 a=CODE(INKEY$(-1))
111 SElect ON a
112 =32:GO TO 3
113 =49:z=1:GO TO 3
114 =50:z=2:GO TO 3
115 =51:z=3:GO TO 3
116 =REMAINDER :piep:GO TO 110
117 CLS:GO TO 3
118 END SElect
119 END DEFine
120 :
121 DEFine PROCedure piep
122 BEEP 2000,3:PAUSE 5:BEEP 2000,6
123 END DEFine

```



Het DILEMMA van computer-kopers...

Meer en minder verwarring en een pluimpje voor quasar.

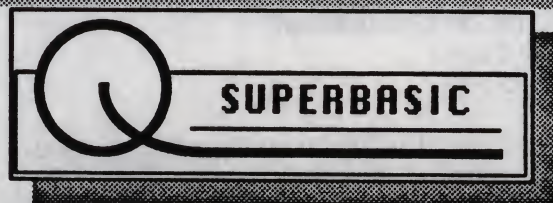
Gezien de vele zinnige informatie die ik de laatste weken uit de QUASAR's gehaald heb, heb ik mij toch maar als lid aangemeld. "Toch maar", daar ik niet erg verenigingsminded ben. Ter introductie: Ik werk nu ca 2 jaar met de QL en hoor tot de mensen die hem vooral als tekstverwerker gebruiken en voor programmeren (in basic). Ik heb hem uitgebreid tot 640k (Miracle), gebruik twee diskdrives (Sinclair), ICE en TOOLKIT 2. Voor diegenen die zich afvragen waarom dan geen ATARI of een kloon (opmerking uit QUASAR):

- a. Toen ik de QL kocht was de ATARI er nog niet.
- b. Toen ik aan uitbreiding begon leverde ATARI nauwelijks zinvolle software bij; ook nu valt dat nog tegen, je moet beginnen er een reeks boeken en programma's bij te kopen voor je er iets mee kunt, en dan is zelfs nude prijs niet zo erg interessant meer.

Verder typt de ST nog slechter dan de QL en is de basic onvergelykbaar met SB. Zelfs nu zou ik eerder aan de AMIGA denken, dan aan de ATARI.c. Voor mensen die op hun werk met een IBM-achtige machine werken kan een kloon interessant zijn, anders nauwelijks. Vooral bij de goedkope kloons ben je volledig op je zelf aangewezen, geen documentatie, geen hulp. Zulke computers verdwijnen meestal al na korte tijd in de kast, om er nooit meer uit te komen. Een ander probleem van de clonen is dat ze niet "compleet" zijn, iets waar de beginner niet op ingesteld is. Hoe kun je je erin verdiepen welke kaarten je nodig hebt, als je eigenlijk niet weet wat je wilt. Hun hele opzet is op zakelijk gebruik gericht. In vergelijking met andere computers wordt de QL met zeer goede documentatie geleverd. Het is mij vanaf het begin een raadsel geweest waarom daar zo lelijk over gedaan werd. Het is een compleet, werkend systeem, een ideale computer om het mee te leren. Valt het "computeren" tegen, dan zijn de verliezen beperkt, zeker tegenwoordig. Het echte probleem is, dat er aspecten aan de QL zijn die je niet meer wilt missen (bv. QUILL, of de leesbare, PASCALachtige basic). Mogelijk kijk ik er teveel vanuit een niet zakelijke wereld tegenaan, maar ik vind MS DOS een ramp en GW basic is een koude douche na SB. Daarom ga ik denken aan een THOR of een FUTURA. Vooral deze laatste heeft mijn belangstelling gezien de (schaarse) gegevens. Ook ik ben benieuwd naar een test. De THOR is mij teveel op zakelijk gebruik gericht, hoewel het een fantastische machine is. Een vergelijkbaar ingerichte goedkope kloon kwam overigens ruim 1000,- duurder uit. Hr Oosterbaan: breid u uw QL dus rustig uit. Ik dacht dat in Engeland iets op de markt was voor uitwisseling van data met MS DOS disks en de ontwikkelingen

bij SANDY doen vermoeden dat dit nog wel verder gaat. Tot besluit: ik ben zeer benieuwd naar een cursus machine taal, ik hoop alleen wel dat het echt voor beginners is, het meeste ervan is voor mij abacadabra. Verder twee vragen: 1. Heeft iemand al ervaring met QL Turbo Quill? 2. Kan iemand mij uitleggen hoe ik iets op een diskette kan vinden. Ik kan wel de naam van een file vinden (de header?), maar hoe haal ik daar de plaats van de file uit. Dat moet toch kunnen. Kennelijk is het besturingssysteem in staat daaruit sektor en track te bepalen, maar hoe? Ik heb een programmatje bijgevoegd om klein getikte directories te maken om bv. op de disks te plakken, een paar mensen hier vinden het erg handig. Het zijn er twee: mini_dir geeft minischift in twee kolommen, mini_dir_u werkt in twee schriftypen en is helemaal instelbaar. Beide versies zijn voor mdv, flp en fdk opgenomen en bedoeld voor een EPSON type printer. Als jullie er wat in zien, dan veel plezier ermee en succes met de vereniging.

... De Ridder, Tonden



Twee maal een mini-directory.

- 1 REMark mini_dir van G.J.G. de Ridder, Tonden. De proc. "direc" is afgeleid van een progr. van van Wolleghem, Kortrijk.
- 2 x=7:y=1:z=2
- 3 CLS:PRINT'\ Dit programma geeft een klein geprinte\ directory van een medium.\ De default-inst. is mdv1 en miniprint,\ voor wijzigen: type m (menu)\ Wijzigen drive: CTRL + SPATIE; dan:\ edit 220, 230, 240 en 270, daarna run.\ Medium niet write-protected!!!\ (progr. schrijft er een file "dirf" op)\ Doe de disk. of cartr. in de drive,\ sluit de printer aan en geef spatie.'
- 4 k=CODE(INKEY\$(-1))
- 5 SELEct ON k
- 6 =109:menu
- 7 =77:menu
- 8 END SELEct
- 9 IF z=1:print1
- 10 IF z=2:print2
- 11 IF z=3:print3
- 12 :
- 13 DEFine PROCedure direc
- 14 DIM dir\$(99,15):DELETE mdv1_dirf
- 15 OPEN_NEW#6,mdv1_dirf
- 16 DIR#6,mdv1_
- 17 PRINT#6,'--'
- 18 CLOSE#6
- 19 OPEN_IN#7,mdv1_dirf
- 20 FOR c=0 TO 99
- 21 INPUT#7,dir\$(c)

En wat zie je na het runnen van dit programma?
Geen basic programma!!

Voeg nu een aantal print regels met een lange tekst en nog wat remarks met tekst toe. Run het nu nog een keer. Nu kom je de geschreven tekst tegen in het begin en even verder op (de variabelen namen zijn niet zichtbaar). Welnu dit is het basic programma! Het basic programma wordt opgeslagen in de vorm van codes, daarbij inbegrepen de variabelen. In het begin van de uitvoer kom je de pas ingetypte tekst tegen met het woord 'run'; dit is de invoer buffer, hier wordt de pas ingetypte tekst tijdelijk bewaard en dan uitgevoerd. Als je nu de uitvoer even door laat gaan komt op een gegeven moment een tabel te voorschijn met alle procedures, functies en op het einde de door ons gebruikte variabelen (en zelfs nog stukjes tekst).

Run het programma nu nog een keer en druk op de ESC-toets precies zodra het programma een variabele afdrukt. De bedoeling is dat je nu het adres van een basic variabele in handen hebt, bijvoorbeeld de o van loop. Deze gaan we nu in een v veranderen door POKE n, CODE('v') in te typen.

Bekijk nu de listing. Zie hier; alles doet het nog alleen de naam van de variabele is veranderd. Dit houdt in dat het gebruik van lange variabele namen de uitvoertijd niet vertraagt, de QL gebruikt deze alleen bij de presentatie aan de programmeur. Het enige wat hem interesseert is de waarde van de variabele welke in een andere tabel staat. Voordat de variabelen te voorschijn kwamen, kwamen eerst de procedures en functies langs welke daar niet voor niets in RAM zitten. Probeer nu eens een letter van een commando te veranderen die in de listing voorkomt, bijvoorbeeld print. Dit weer op de zelfde manier als hiervoor en vraag weer een listing op. En zie hier; ook dit verandert hij terwijl hij het oude commando niet meer herkent; er is echt iets veranderd.

Met dit gegeven in de hand kunnen we nu een nederlandstalige basic maken met als enige beperkingen dat de lengte gelijk moet blijven en dat het commando in die lijst aanwezig moet zijn (def proc komt er niet in voor, dit onder andere vanwege zijn afkortmogelijkheid). Je kan bijvoorbeeld de volgende dingen veranderen read wordt lees, call wordt roep en peek wordt zoek. Na het eerste enthousiasme komen al snel de verdere beperkingen boven water; op de een of andere manier accepteert hij geen leestekens in een commando dat veranderd wordt. Hij drukt het nog wel af bij een listing maar kent het niet meer bij de invoer. Toch doet hij tijdens de uitvoer van het programma met een veranderd commando met leestekens of hij deze wel kent.

Nu kan het dus voorkomen dat als je zo een commando met edit ertussen voegt hij dit niet kent, terwijl hij het andere identieke commando wel kent (hij ziet de laatst toegevoegde als een gewone basic procedure aanroep).

Een andere beperking is de commandolijst. Hier komen uitsluitend eenvoudige machinetaal aanroepen in voor. Die commando's die het programmaverloop bepalen for, repeat, exit, if etc zijn hier afwezig. Deze zitten veilig in ROM opgeborgen.

Nu weer terug naar de basic pointer. Je moet basic vergelijken met een stapel boeken die je boven op elkaar legt. Nu wil je dat deze stapel boeken altijd tot het plafond reikt zodat je de onder liggende ruimte maximaal kan benutten.

Dit houdt in dat iedere keer als je er een boek afhaalt of er bij legt, je steeds de boekenplank moet verstellen (zie je het al voor je?). Dit is nu precies de manier waarop QDOS het basic programma met bijbehorende zaken behandelt. Waarbij de rol van boeken plank wordt vervuld door de basicpointer register A6. Als je nu in het basicprogramma iets tussenvoegt wordt A6 verlaagd en de hele zaak wordt opgeschoven. Ook als het DIM statement wordt uitgevoerd moet er ruimte gereserveerd worden in het basic gebied, dus weer wordt A6 verlaagd en de hele zaak opgeschoven. Dit is precies de reden waarom het DIM statement tijd kost.

Nu geloof je dit natuurlijk maar half, maar geen nood! Verander het basic programma zo, dat het alleen de pointer afdrukt. Voer nu het programma uit zonder en met het DIM statement, zonder hierbij een tweede keer RESPR te gebruiken! De informatie die in het basicgebied staat wordt geschreven van boven naar beneden waardoor niet alles opgeschoven moet worden als je een regel toevoegt (hetgeen bij het laden gebeurt), zodat alleen de basic pointer verlaagd moet worden. FOUT!

Onderin het basicgebied zit precies een kwart kilobyte met basicgegevens. Dit kleine gedeelte verhuist dus wel steeds mee. Die basic gegevens zijn onder andere de regel nummer die uitgevoerd wordt, waar de reken stack is en waar het laatst gelezen data statement staat.

Er is nog zoveel meer...

Boven Superbasic in het geheugen zitten nog twee gebieden; het vaste proceduregebied en het voorbijgaande programmeergebied.

Van het eerste kan je een stukje reserveren met het basiccommando RESPR(stukje) waarin je dan machinetaal routines kwijt kunt voor basic. In het tweede komen de 'Jobs', er in gebracht met EXEC(W). Voor deze jobs wordt eerst een stuk geheugen gereserveerd en de job wordt ingeladen, waarna deze een eigen leven gaat leiden. Jobs kan je niet verschuiven (althans niet binnen QDOS).

Dit houdt in dat als er een job in het geheugen zit, de RESPR functie het niet meer doet. Bij het verwijderen van een job wordt het door hem


```

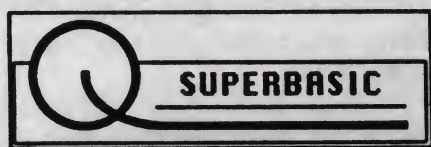
1 REMark Mini_dir door G.J.G. de Ridder, Tonden;
  proc. "direc" naar van Wolleghem, Kortrijk.
2 CLS:PRINT'\ Dit programma geeft een klein
  geprinte\ directory van een cartridge.\ Cartridge
  niet write-protected!!!\ (progr. schrijft er een file
  "dirf" op)\ Doe de cartridge in de drive,\ sluit de
  printer aan en geef spatie.'
3 k$=INKEY$(-1):printen
4 DEFine PROCedure direc
5 DIM dir$(99,15):DELETE mdv1_dirf
6 OPEN_NEW#6,mdv1_dirf
7 DIR#6,mdv1_
8 PRINT#6,'---'
9 CLOSE#6
10 OPEN_IN#7,'mdv1_dirf
11 FOR c=0 TO 99
12 INPUT#7,dir$(c)
13 IF dir$(c)='---':EXIT c
14 END FOR c
15 CLOSE#7

```

```

16 c=c-1
17 END DEFine
18 DEFine PROCedure printen
19 OPEN#3,ser1
20 BAUD 9600
21
  PRINT#3,CHR$(27);CHR$(33);CHR$(4);CHR$(2
  7);CHR$(83);CHR$(0);CHR$(27);CHR$(65);CHR
  $(6);
22 direc
23 PRINT#3,'DISK: 'dir$(0);' dir$(1);' vrij\"
24 FOR n=1 TO c-1 STEP 2
25 IF LEN(n)=1:PRINT#3,' ';
26 PRINT#3,n!dir$(n+1);:a=20-
  LEN(dir$(n+1)):PRINT#3,FILL$( 'a);
27 IF LEN(n+1)=1:PRINT#3,' ';
28 PRINT#3,n+1!dir$(n+2)
29 IF dir$(n)='---':EXIT n
30 END FOR n
31 END DEFine :PRINT#3,"\\":GO TO 3

```



Op safari door de QL

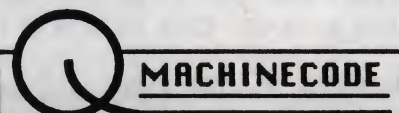
Voor velen is de QL met recht een black box.
Maarten doet een poging uw computer iets
transparanter te maken.

Met behulp van het toetsenbord kan je toch leuke ontdekkingsreisjes maken door de QL, waarbij je steeds weer verbazingwekkende zaken tegen komt. Het eerste wat voor zo'n ontdekkingsreisje in aanmerking komt is Superbasic, hetgeen je al direct problemen geeft, want waar zit Superbasic eigenlijk?

Het is namelijk zo dat Superbasic zich geheel aanpast aan andere in het geheugen geplaatste programma's en daar gaat zitten waar er nog ruimte is. Gelukkig heeft hij een adresje achter gelaten, dit adresje zit in een van de registers van de 68000; A6.

Nu is een register niet vanuit basic met het peek-commando op te vragen. De oplossing is een kleine routine die de inhoud van A6 verplaatst naar een geheugen adres zodat dit wel met peek is op te vragen. Deze routine ziet er als volgt uit:

43FA			
0008	start	lea.l	pointer,A1
228E	move.l	A6,(A1)	
7000	moveq	#0,D0	
4E75	rts		
0000			
0000 pointer	dc.l	0	



```

10110101101000101010001010
10101010000111101010011001
10110001100111110100110101
01011110001111000110001101
10101010101011111000111000

```

En dit in een basic programma dat zichzelf gaat bekijken:

```

mode 4 : RESTORE
routine = RESPR (16)
FOR n = 0 to 13
  READ x
  POKE routine+n,x
END FOR n
CALL routine
pointer = 0
FOR n = 1 to 4
  pointer = pointer*256 + PEEK (routine+9+n)
END FOR n  n = pointer
REPEAT loop
  PRINT CHR$ (PEEK (n));
  n = n + 1
  IF KEYROW (1) = 8 THEN EXIT loop
END REPEAT loop
DATA 67,250,0,8,34,142,112,0,78,117,0,0,0,0

```


gebruikte geheugen weer vrijgegeven.

Het bovenstaande houdt in dat bij die handelingen iedere keer de basic wordt verschoven. Dit geeft de speciale plaats van Superbasic binnen de QL al aan. En niet alleen daarom, er zijn binnen QDOS speciale routines aanwezig die het mogelijk maken dat basic meer geheugen krijgt en dit later bij NEW of CLEAR weer vrijgeeft. Daarom duurt het even als je na CLEAR de cursor weer terug krijgt.

En verder is het mogelijk de syntax uit te breiden van de basic {waardoor er een enorme hoeveelheid toolkits op de markt zijn}. Nog iets speciaals voor basic is de CTRL-spatie toetscombinatie; ook al is een andere job op het ogenblik dat je deze toetsen indrukt actief, toch krijg je 'not complete' in het basic invoer scherm te zien.

En nu iets heel anders...

Aan de hand van twee programma's wilde ik laten zien hoe de basic interpreter tegen basic procedures aankijkt met locale variabelen. In de vorige Quaser schreef Marc over machinetaal waarbij hij uitlegde hoe een stack in elkaar zit. Lees dit toch eens door ook al vind je machinetaal (nog) te moeilijk.

Een stack is namelijk van het grootste belang voor het hele gebeuren binnen een computer. Basic gebruikt er twee, één voor rekenkundige bewerkingen en één voor programma-verloop gegevens.

De eerste heeft de zelfde functie als de stack waar Marc het over had en op de tweede gaan bijvoorbeeld bij het springen naar een subroutine de plaats waar deze vandaan sprong (en dus de plaats waar hij straks zijn weg moet vervolgen).

Het gebruik van dit soort dingen wordt pas leuk bij recursie. Recursie is het zichzelf aanroepen van een procedure binnen de procedure. Hier is een programma dat niets anders doet dan eerst optellend een aantal getallen op de stack zetten en ze even later er weer af te halen en af te drukken.

```
100 stack = RESPR(99) : stack= stack + 99
110 max = 10
120 n = 1 : GOSUB 130 : STOP
130                                     REMark begin
sub
140 POKE stack,n
150 stack = stack - 1
160 IF n < max THEN n = n + 1 : GOSUB 130
170 stack = stack + 1
180 n = PEEK (stack)
190 PRINT n
200 RETurn
```

Zoals je ziet zet je niet iets op de stack maar onder de stack. Gelukkig heeft de QL een Super basic zodat dit programma veel mooier kan waarbij regel 140,150, 170 en 180 door de interpreter verzorgd worden. Dit gaat er dan als volgt uit zien.

```
max = 10
loop 1
DEFine PROCedure loop (n)
  LOCAl v
  v = n
  IF n < max then v = v + 1 : loop v
  PRINT n
END DEFine loop
```

Hetgeen hetzelfde resultaat heeft {alleen ietsje sneller}.

De presentatie van een getal. Diegene die het ellipsprogramma van Marc bekeken hebben zullen wel ontdekt hebben dat er iets heel raars aan de hand is met de getallen buiten basic. QDOS werkt namelijk uitsluitend met floating point getallen. Dit heeft voor basic het voordeel dat het uitermate eenvoudig wordt om programma's te schrijven waarbij getallen bewerkt worden; je harkt gewoon hele en floating point getallen door elkaar. Het is alleen vervelend dat QDOS van je verwacht dat je ook bij het aanroepen van bijvoorbeeld de cirkelroutine in machinetaal de getallen eerst om zet in floating pointgetallen, terwijl dit eigenlijk niet nodig is. Een bijkomend nadeel is dat de basic er niet sneller op wordt.

Als laatste nog een paar opmerkingen.

- De QL heeft een grote hoeveelheid systeemvariabelen {5K} die in het tweede scherm staan. Deze kan je zichtbaar maken door het intypen van POKE 98403,128. En als je een tijdje bezig bent geweest {verschillende programma's geladen enzo} staat er meestal nog veel meer in dat scherm.
- Het DATA statement is erg flexibel. Je kan er niet alleen tekst en getallen in kwijt maar ook hele expressies. Zo kan je er goniometrische functies inzetten die je als een soort define function kunt gebruiken.

Als voorbeeld:

```
FOR n = 0 TO 90 STEP 10
  RESTORE : x = RAD (n)
  READ y :PRINT y
END FOR n
DATA COS(x)*SIN(x)
```

- Als naslagwerk gebruik ik het boek 'Advanced user guide' van A.C. Dickens. Een boek dat ik iedereen kan aanbevelen.
- Dit verhaal is verre van volledig, de QL zit veel ingewikkelder in elkaar!

Maarten de Moor

P.S. Waarom geen bijeenkomst op zondag

(Waarom wel? @)

NIEUWS UIT HILVERSUM

Jan Bredenbeek.

Eindelijk heb ik mezelf ertoe kunnen bewegen om achter QUILL te gaan zitten en een stukje te schrijven voor de Quasar.

Zoals een aantal van jullie inmiddels weten heb ik mezelf opgezadeld met de taak het programma te schrijven voor de toekomstige QL-databank. Tegen de tijd dat jullie dit lezen ben ik misschien al bezig met de eerste testen. Ik wil al vast een tipje van de sluier oplichten.

We beginnen met een zogenaamd Bulletin Board Systeem, waarin een aantal mededelingen staan en waaruit je files kunt downloaden. Het ligt in de bedoeling om het systeem dan geleidelijk uit te breiden met een mailbox (berichtendienst) e.d. Er wordt voorlopig alleen gewerkt op 1200/75 baud omdat vrijwel ieder modem die snelheid heeft. Het downloaden zal voorlopig alleen met het XMODEM-protocol gaan.

Overigens heb ik ook een terminalprogramma ontwikkeld dat volgens dit protocol werkt en dat binnenkort via de cartridge-service beschikbaar komt.

De databank zelf komt bij de Spanjers te staan en is via een eigen telefoonlijn 24 uur per dag beschikbaar (in principe).

Voor alle duidelijkheid: als het zover is melden we het in de Quasar.

Momenteel werken de Spanjers nog aan de hardware en ik aan de software.

In het aprilnummer zag ik een stukje van Frank Troost over QL-bestanden in databanken. Zelf onderhoud ik ook onder de naam SINCLAIR INFO een dergelijk bestand in twee databanken, t.w.:

VieWorld, tel. 035-12633 of 231161 (na 18.00 uur). Viewdata mode, log in met het gastnummer (222222-2222-2222) en kies dan pagina *70#.

Hobbytel, tel. 035-231513, 24 uur per dag op 4 lijnen.

Gastgebruikers beperkt toegang, volledige krijgt u door abonnee te worden (f. 37,50 per jaar). Kijk op pagina *290# (viewdata mode).

Verder heeft de Fido-node in Kampen (05202-24380) een message- en filegebied voor Sinclair-computers.

Wat de opmerkingen van Ard Jonker in het aprilnummer betreft over slecht gestructureerde SuperBasic programma's moet ik hem gelijk geven. Wel

is het zo dat een hoop slecht programmeerwerk in de hand gewerkt wordt door de slechte User Guide. Wat dat betreft kan "The Definitive Handbook" van Jan Jones wonderen doen.

Niet alleen SuperBasic programmeurs maken "verkeerde" programma's, de machinetaalkrassers kunnen er ook wat van. Neem bijvoorbeeld het ellips-programma van Marc Kool op pagina 460.....

Beste Marc, als je wat ruimte nodig hebt op de RI-stack roep je gewoon de BV_CHRIX routine aan (vector \$11A) met in D1 het aantal bytes. Wat jij doet (de stack in de common heap zetten) is helemaal fout, omdat je bij SuperBasic register A6 NOOIT mag veranderen, ook niet "even" op de stack zetten! Wordt er gedurende jouw routine nl. een job gemaakt of verwijderd, dan verandert de SuperBasic A6 waarde en crasht jouw routine.

Verder is absolute adressering bij SuperBasic (wat ik soms ook tegenkom bij een aantal programmeurs) volledig uit den boze omdat het hele SuperBasic gebied van de ene op de andere instructie kan verschuiven.

Het bovenstaande verhaal is misschien een beetje te "freakerig" voor publikatie in de Quasar maar ik wil het toch even kwijt.

Tot mijn genoegen heb ik kunnen constateren dat Ard Jonker de ondankbare taak van machinecode-cursusleider op zich heeft genomen. Zal machinecode nu dan eindelijk uit het verdomhoekje komen? Afwachten maar. In ieder geval heb ik mijn bevindingen over de GST Macro Assembler en twee Elektoer boekjes ook op papier gezet elders in deze Quasar.

Voor het geval dat u nog een machinecode monitor zoekt: Voor slechts drie gulden kunt u er een in uw bezit krijgen via de cartridge service (Quasar C07, MULTIMON).

Eind april heb ik versie 2.0 naar Kees gestuurd. Deze versie is t.o.v. de vorige (1.0 van een jaar geleden) aanzienlijk verbeterd en uitgebreid. Je kunt nu ook vanuit de monitor een _EXEC programma inladen, de in de QL aanwezige jobs bekijken enzovoort. De handleiding is nu 25K groot en als _doc file bijgevoegd.

Jan Bredenbeek.

DE ELEKTUUR 68000-REEKS

Het is nu zo'n jaar geleden dat uitgeverij Elektuur met twee boeken op de proppen kwam, waarin zowel de hardware- als de software-aspecten van de 68000- en 68008-microprocessor belicht worden. Beide delen zijn geschreven door L. Nachtmann, onder meer bekend van de Elektuur Junior computer.

In deel 1 wordt het volledige hardware-gebeuren van de 68000 uit de doeken gedaan. Niet alleen worden de 68000 hardware-kenmerken behandeld maar ook de aansluiting op diverse 68XX- en 68XXX peripheral chips. In een ca. 40 pagina's tellende appendix wordt bovendien de 68008 besproken, en wat de verschillen zijn met de 68000.

In deel 2 wordt de instructieset van de 68000 (welke dezelfde is als die van de 68008) besproken. Nogal overdreven hierbij vind ik dat men van de Bcc, DBcc en Scc instructies alle 16 mogelijke varianten apart bespreekt terwijl deze zonder al te veel verlies aan duidelijkheid samen genomen hadden kunnen worden. Bovendien bevat deel 2 - en dat mis ik zo vaak - een overzicht van de kloktijden van alle instructies, zowel voor de 68000 als 68008.

Vergelijk de kloktijden van de 68008 met die van de 68000 en je weet meteen waarom de QL niet zo snel is als hij zou moeten zijn.

Hoewel ik het zelf niet heb kunnen verifiëren zijn beide delen voor het grootste gedeelte rechtstreeks overgenomen van de Motorola 68000-manual, volgens enkele mede-clubleden die ze ook gelezen hebben.

Nu hoeft dat niet direkt een nadeel te zijn want beide delen staan tenminste wel boordevol met belangrijke en waardevolle informatie. Bij een aantal computerboeken laat men nog wel eens belangrijke aspecten achterwege om beginners niet te veel af te schrikken, hier is daar gelukkig geen sprake van.

Een groter nadeel is dat deze serie niet specifiek voor de QL is geschreven. Wie niets met hardware te maken heeft kan in principe volstaan met alleen deel 2 maar zal dan toch nog wel een boek met een beschrijving van het QDOS operating system moeten kopen.

Verder is deze serie niet op te vatten als een cursus machinetaal, dus wie dat nog moet leren dient er nogmaals een boek bij te kopen. Dat gaat dus aardig in de papieren lopen, maar het voordeel

is natuurlijk wel dat je dan de beschikking hebt over een hoop informatie.

Al met al is deze serie dus bedoeld voor de wat gevorderde 68000(8) programmeur, en zeker aan te raden voor de hardware-knutselaars.

Jan Bredenbeek.

68000, anatomie van een super-micro-processor, deel 1 en 2.

Uitgeversmaatschappij Elektuur, Beek (L).

Deel 1 (232 pagina's):

ISBN 90-70160-41-2

Deel 2 (254 pagina's):

ISBN 90-70160-42-2

DE GST MACRO ASSEMBLER EN LINKER GETEST

De hier besproken assembler/editor/linker wordt aangeboden als de officiële Sinclair Assembler. De assembler en linker zijn van GST afkomstig, de editor is de bekende "ed" van Metacomco. Gezien de bekendheid van de editor zal ik daaraan weinig aandacht besteden. Het is een goede editor met veel mogelijkheden, al zijn sommige commando's zoals "block delete" en "insert file" ontzettend traag omdat ze regel voor regel alles op het scherm laten zien. Vreemd genoeg is "block insert" wel snel. Het wegschrijven van de tekst naar de file kan volgens mij ook wel wat sneller.

Dan de MACRO ASSEMBLER. Dat schrijf ik expres met hoofdletters omdat het de beste en meest uitgebreide assembler is die ik tot nog toe voor de QL ben tegengekomen. De macro-faciliteiten zijn zeer uitgebreid; zowel parameters als lokale variabelen zijn mogelijk; verder lokale labels in macro's waarnaar de assembler d.m.v. een GOTO label-opdracht kan springen (ook terug in de tekst!). Variabelen zijn in essentie string; door de naam tussen blokhaken te zetten krijg je een substitutie in de tekst. Daarmee zijn zeer fraaie constructies mogelijk zoals het automatisch genereren van instructies en labels en simulatie van arrays. Verder zijn een aantal functies voorhanden; o.m. een "macro call counter" zodat je verschillende labels kunt maken bij opeenvolgende macro-aanroepen. Conditionele assembly is uiteraard ook mogelijk.

Het nut van al deze faciliteiten wordt gedemonstreerd door de bijgeleverde "macro library". Deze

bevat een aantal macro's waarmee gestructureerd programmeren in assembler mogelijk wordt. Deze structuren zijn:

IF-THEN-ELSE, REPEAT-UNTIL, WHILE, FOR en CASE (met default).

Voorbeelden:

```
IF D1 GE.B #'a' AND D1 LE.B #'z' THEN
SUB.B #'a'-'A', D1 ; maak D1 hoofdletter
MOVEQ #0, D2 ; geef aan "kleine letter"
ELSE MOVEQ #1, D2 ; geef aan "hoofdletter"
ENDIF
FOR.W D0 =#1 TO #10 DO MOVE.W D0, (A0)+
ENDFOR
REPEAT ADDQ.L #1, D0
UNTIL EQ
```

Overigens testen deze macro's zelf of ze goed genest zijn en geven zonodig een foutmelding! De assembler zelf is een "slimme" assembler; een BRANCH terug wordt automatisch "kort" of "lang" genomen; bij een voorwaartse branch die lang is maar kort kan zijn wordt een waarschuwing gegeven. Ook worden dingen als MOVE.L OFFSET(A6), A0 waarbij OFFSET=0 automatisch veranderd in MOVE.L (A6), A0.

Dan komen we bij de linker.

Deze zul je nodig hebben als je bijv. een groot programma in stukken wilt splitsen of een library van veelgebruikte routines wilt maken. Je kunt labels en symbolen in een module d.m.v. XDEF in assembler toegankelijk maken voor andere modules. Deze moeten dan een XREF declaratie hebben voor het betreffende symbool. De linker zorgt er dan voor dat de modules keurig aan elkaar gelinkt worden en alle onderlinge verwijzingen kloppen. Overigens hoeven die verwijzingen niet alleen in de vorm "JSR routine" te zijn; iets als "DC.W routine-'" wordt ook correct gelinkt. Absolute symbolen geef je met XREF.S of XREF.L aan; zoals bv. symbolen die je gebruikt als offset van een register.

De assembler en linker kennen ook SECTION's; dit zijn gebieden waarin code geassembleerd wordt en welke via een naam onderscheiden worden. Je kunt bv. aparte gebieden maken voor code en meldingen zodat debuggen eenvoudiger wordt. Wil je bv. een melding in de assembler-tekst achter een routine plaatsen maar welke in werkelijkheid veel verder weg is tussen andere meldingen dan doe je dat met SECTION MELDINGEN; DC.B "melding". Met SECTION CODE schakel je dan weer terug naar het code-gebied.

Zowel de assembler en linker kun je starten met een "command line" als je de Toolkit hebt. Beiden geven een uitgebreide listing naar een aangegeven file; onder meer wordt hierbij een "cross reference" van labels en symbolen gegeven.

Deze listing kan geheel of gedeeltelijk worden uitgeschakeld. De handleiding bestaat uit een A5-ringband en is ruim 90 pagina's voor de assembler, 50 voor de linker en 15 voor de editor. Zeer uitgebreid dus.

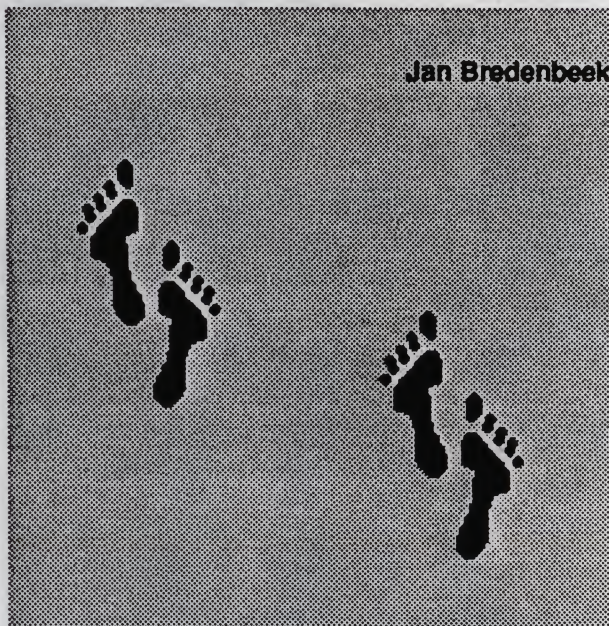
Ik heb deze assembler vergeleken met de Metacomco assembler waar ik een tijdje mee gewerkt heb. Mijn conclusie is dat deze assembler met kop en schouders boven de Metacomco uitsteekt. Niet alleen zijn er veel meer mogelijkheden; de assembler is ondanks dat veel compacter. De Metacomco is een log programma met 3 overlays welke traag van Microdrive gelezen worden. De GST laadt in één keer in. Verder is de grootte van het werkgeheugen bij de Metacomco beperkt; de GST past zijn werkruimte dynamisch aan. Daarom kan de GST zeer lange files moeiteloos aan. Bovendien is de GST sneller.

De link-mogelijkheden bij de Metacomco zijn beperkt; alleen eenvoudige labels welke ook nog via een speciale linker gelinkt moeten worden.

Voor mij was de keuze dus niet moeilijk: Weg met de Metacomco en verder werken met deze assembler, en ik denk dat vele gevorderden na het lezen van bovenstaand verhaal het met me eens zijn. Hoewel deze assembler vanwege zijn complexiteit misschien niet de meest ideale is voor beginners, raad ik ook deze groep aan met deze assembler aan de slag te gaan in plaats van met een eenvoudige waarmee men snel tegen de beperkingen aanloopt.

De volledige macro-versie (ca. 30K) wordt bij bepaalde Funtronics-filialen aangeboden voor rond de 160 gulden. Bij de GST C-compiler wordt ook een uitgeklete versie zonder macro's geleverd (ca. 25K) en tenslotte bestaat er dan nog een nog meer uitgeklete versie van 18K welke geen macro's en geen link-faciliteiten kent, maar voor kleine programma's toch te prefereren valt boven de Metacomco.

Jan Bredenbeek



ingezonden mededelingen



Aangeboden; i.v.m. overstap naar ander systeem:

QL (128K, basispakket incl. software) f325,-/24 cartridges met enige software in box. f 150,-/1 RS232-kabel voor QL f25,-/1 joystick verloopkabel f15,-/1 IC 23 ZX8302 voor QL f25,-/1 monitorkabel (composite) f15,-/1 sinclairgids 2 en 3, QL world jan en feb 87 f10,-

In één koop f500,-

Voor de kenner: 1 Philips diskdrive X 3113 SS 80 tr 360 Kb. f100,- en 1 Samsung monitor/tv 8 voork stat/ 31 cm f150,-.

Jan Rademaker 01829 - 3972.

Naar onze mening mag een tweedehands QL op dit moment niet meer kosten dan de winkelprijs.... en dat zou neerkomen op censuur
Maar wie zijn wij?@

Door een vergeetachtigheid van mij is er de vorige maal een kleine advertentie niet geplaatst. Hier volgt hij dan:

TE KOOP:

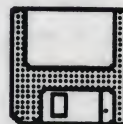
QL 512 K - QL-printer - Joystick

Diskdrive 720 Kb - Monitor

Boeken - Software.

f2300,-

De Heer Dikmans: 02269 - 3236



Het programma bij het vorige maand verschenen artikel over formattering van QL en ATARI-diskettes zal binnenkort in de cartridge/disk-service te verkrijgen zijn. Het programma is bij nader inzien te groot en te veel gericht op een naar alle waarschijnlijkheid te kleine groep leden, dat we er beter aan dachten te doen de ruimte voor andere artikelen te gebruiken.....

Aangeboden:

Printer: Brother EP 22 met kabel. Schrijfmachine: Canon s-60. Lege QL-kast, i.v.m. ombouw naar Sandy. Voeding 2 floppy-drives, i.v.m. ombouw naar Sandy tegen elk aannemelijk bod. Hans Snel, Postbus 61289, 1005 HG Amsterdam. 020 - 880126.



Laat, later, laatst....te laat?

Uit wat oudere papieren die ik van John ontvang, haalde ik nog het vorige verzoekje. Hopelijk niet te laat om nog het vermelden waard te zijn...

februari 1987,

In het blad Personal Computer World van februari 1986 stond een listing in MBasic voor een eenvoudig 'expert systeem'. Sedert die tijd

heb ik verschillende keren geprobeerd die listing voor de QL te vertalen. Maar tot op heden zonder succes.

Is het jou wel gelukt? Dan zou ik het erg op prijs stellen als ik een seintje van je krijg. Is het je niet gelukt, maat ben je het nog steeds aan het proberen? Misschien komen we samen verder. Laat eens wat van je horen op telefoonnummer 04130 - 44048 (na 19.00 uur)

KOPIJ, en HOE 1?...

Sorry beste mensen. Ik heb het al enkele malen eerder verzocht, gemeld, verteld en gezegd; stuur je kopij op een cartridge en niet op papier of floppy. Beide media bezorgen mij hoofdbrekens. Moet ik dan toch maar als een gek artikelen van papier over gaan zitten tikken, moet ik dan de straat op met een floppy om iemand te vinden die het even op cartridge wil zetten.....

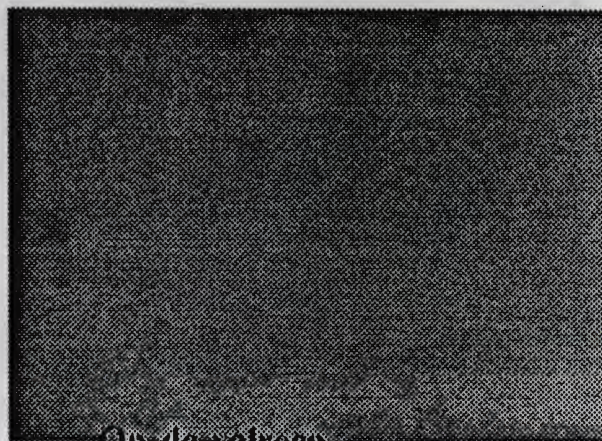
Nogmaals, ik heb de handen vol aan het in elkaar zetten van de Quasar. Lever alsjeblieft je kopij in op cartridge. Iedereen heeft toch twee microdrives ingebouwd in zijn QL? En als je floppies gebruikt kun je een cartridge zeker wel even missen.

Echt, op een andere manier ingezonden dan per cartridge, gaat de kopij retour. De vraag is dan of je het weer op tijd bij mij terug krijgt om het nog in de Quasar te krijgen die je in gedachte had; vooral als je kopij op het nippertje binnen is, kan dat wel eens niet meer lukken.

Als het even mogelijk is, zend dan ook altijd een uitdraai van je kopij mee. Dit heeft twee redenen. Ten eerste kan ik dan vast doorlezen en op onderwerp sorteren, en ten tweede:

Mocht er iets niet helemaal duidelijk overekomen tijdens het overseinen naar de Mac, tabstops worden vaak niet gebruikt door jullie, dan is het voor mij moeilijk om de oorspronkelijke vorm weer terug te vinden. Denk aan met Quill gemaakte tabelletjes en aan MC-listings. Graag dus een uitdraai erbij.

RON.®



Op de valreep.....

Helaas het π _doc kreeg ik niet ingelezen, waarschijnlijk om dat bij de copy mdv1_ π _doc to ser1i opdracht die ik gebruik een teken als π niet toegestaan is. In Quill misschien wel, maar die had ik tijdens het leggen van de laatste hand aan deze Quasar niet bij me. Volgende keer.....

®Ron